Multi-layered model of individual HIV infection progression and mechanisms of phenotypical expression

Dimitri Perrin

A thesis

submitted in partial fulfillment

of the requirements for the degree

of

Doctor of Philosophy

in the School of Computing, Dublin City University,

Dublin City University

Faculty of Engineering and Computing, School of Computing

Supervisors: Prof. Heather J. Ruskin, Dr. Martin Crane

September 17, 2008

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: .....

ID No.: .....

Date: .....

Contents

Abstract

Development of a computational model is reported, which focuses on aspects of individuality in biological phenomena. Of particular interest are factors related to the immune response to HIV infection and how these interact within a complex system leading to significant variations in latency period. A *multi-layered* approach is chosen, with the most crucial layer comprising a large-scale agent-based model lymph network. Agent-cell reciprocity permits isolation of key features, e.g. cell mobility, viral mutation. Cell mobility, in particular, is incorporated in an innovative way, due to parallel implementation which permits realistic size of the lymph network. This implementation, using lymph nodes as key structural units, also permits inclusion of localised effects, e.g. early infection in the *gastro-intestinal tract*, which is increasingly reported as having a decisive impact on long-term disease progression.

Additional layers of the model provide a foundation for investigation into the phenotypical-level mechanisms of the immune response implemented in the main layer. The objective here is to gain insight on how these mechanisms originate from a sub-layer of interactions, (both genetic and epigenetic). Gene interactions are commonly studied through microarray techniques and generate highly complex dependencies. A validation framework for such analysis is proposed, and a model of infection-induced epigenetic changes, (within the agent-based paradigm), is developed.

Acknowledgements

Foremost, I would like to thank Prof. Heather J. Ruskin and Dr. Martin Crane for their support, expertise and research insight. Their thoughtful advice helped me maintain a sense of direction during the course of my work.

I also gratefully acknowledge support from the Irish Research Council for Science, Engineering and Technology, through the Embark Initiative. This work would not have been possible without them.

I want to express my gratitude to the whole research group, for the excellent atmosphere and for their stimulating discussions, their help with technical issues and general advice.

Collaborative work is ubiquitous in research, and often is at the origin of very useful work. This is confirmed in this thesis, and the colleagues involved also made a very enjoyable human experience. Gráinne Kerr, Christophe Duhamel and Toshikazu Ushijima deserve my gratitude for this.

Life as a researcher can sometimes be difficult and, even though I enjoyed most of it, there has some tough times too. I am very grateful for the love and support of my wife Vanessa, who was always there for me, be it to share my enthusiasm for research or to help me get through more stressful periods when needed. To her I dedicate this thesis.

I would also like to express my gratitude to my family, (including Vanessa's), who supported me over the years, even long before I started this PhD. Support from friends has also been essential. I will not turn this into an exhaustive list, but a few deserve to be mentioned here: warm thanks to Alain, Chris, Didier, Jean-Marc and Shiho.

List of Tables

List of Figures

Chapter 1

Introduction

1.1 HIV infection progression

Immunity in the human body is obtained through emerging properties of a very complex system. It involves a multitude of cells and organs, with very specific functions and numerous possible interactions. This complexity often hinders understanding of the range and variety of immune responses. Large-scale effects are easily observed, and microscopic studies give a better insight into the sequence of interactions, but links between these two levels are difficult to establish, in particular if we are looking for a quantitative description. The situation is pronounced with respect to HIV infection. Mechanisms by which an HIV virion infects an immune cell, and its genetic material is incorporated into the host chromosome, are known, as are processes leading to production and liberation of new virions. Macroscopic progression from initial HIV infection to AIDS onset are equally well described.

However, there are still millions of people living with HIV (UNAIDS, 2004), and the process by which interactions between the immune system and HIV lead to such variability in individual experience of infection has yet to be fully described. Development of a vaccine is even further down the road (Garber et al., 2004), although recent therapeutic efforts have led to better control of disease progression (Sterne et al., 2005).

To facilitate analysis of this complex system, numerous *in silico* models have been developed, (see e.g. Celada and Seiden (1992)). Early efforts suffered from a relative lack of biomedical data, and from limited computing resources then available. However, a number of these efforts and of subsequent developments, (see e.g. Pandey et al. (2000); Bernaschi and Castiglione (2001), were able to match some signatures of HIV, serving as a *proof of concept* and ensuring continued interest and ongoing efforts in the field of computational immunology.

Recent models are, of course, more refined (see e.g. Ruskin and Burns (2006)), employing more sophisticated approaches and computer resources, and offering valuable insights into specific aspects of the system, despite their limitations.

1.2 Mechanisms of phenotypical expression

A *phenotype* is any observable characteristic of an organism, such as biochemical properties, morphology, behaviour. Immune response to HIV infection is, therefore, a phenotypical event: it involves interactions between observable physical or biochemical characteristics of the organism, i.e. immune organs[1] and cells[2], and the virus. Indications are that individual variations in the length of the latency period may be attributed to cell-level characteristics, such as *cell mobility* and *viral mutation*.

A complex system regulates how the complete set of genes of an organism, the *genotype*, is expressed and results in these phenotypical characteristics. In the long term, a better understanding of this system is needed for a more accurate analysis of the individual characteristics, and their influence.

The all-genetic paradigm introduced in the early days of Genetics has been abandoned: the phenotype is not a deterministic consequence of the genotype. The first amendment to this was to consider environmental effects. Recent work has led to the inclusion of another

[1] This refers to all organs responsible for immunity, from producing cells, (main function of the thymus), to staging immune response, (main function of the lymph nodes). These organs are detailed in Section 2.1.

[2] These include lymphocyte lineages, macrophages and other cells involved in the immune response, as detailed in 2.1.

parameter in the equation, *epigenetic changes*: modifications of the chromatin structure, heritable and leading to altered gene expression. An introduction to this paradigm shift can be found in Speybroeck (2000).

This aspect has, so far, been neglected in development of *in silico* biological models at all levels. Accounting for this in a *multi-layered model* of the immune system offers great potential. In the broader context of Computational Biology, this would be a decisive breakthrough for a wide range of biological systems, from cancer initiation to neural development.

1.3 Objectives and challenges

A model is typically selective, and cannot include all parameters and interactions. The advent of large-scale, parallel computing, however, offers opportunity for increased refinement of biological models. In particular, a better understanding of the events behind emergence of individuality is of special interest.

Further refinement can be obtained through a multi-layer approach: when a single model can not include some parameters or interactions, a complementary model may be developed, which focuses on these specific features. The limited scope of this second model ensures a more detailed representation of these. Results can then be integrated into the main model. Such an approach is taken here.

The main layer considers the immune response to HIV at the phenotypical level. If the basics of the infection and of the immune response are now better understood, (see Chapter 2 for details), it is, nevertheless, still largely unclear how these cell-level events interact with each other to lead to the HIV-characteristic disease progression.

Most previous modelling efforts, because of limited computing power or implicit modelling choices, failed to integrate what can be considered a key element of the immune response: cell mobility. Using a large-scale agent-based model, a key objective is to realistically account for this.

Recent research, (see e.g. Mattapallil et al. (2005)), highlights another feature of the body-wise progression of the disease: there are localised effects, such as those within the gastro-intestinal tract, that require incorporation into computer-based models. The proposed approach, using lymph nodes as a key model element, permits inclusion of this layer of the system.

Since phenotypical characteristics are the results of a complex system, interactions involved can not be directly included in this large-scale agent-based model and a second layer is, therefore, required. This layer is divided into two additional models which complement the main model and focus, respectively, on genetic and epigenetic considerations.

The first one is developed to analyse microarrays. To better understand *gene expression*[3], microarray technologies are extensively used. Yet, there is no clear consensus on analysis techniques. Here, through adaptation of optimization techniques to microarray biclustering, and introduction of properties to assess associated weighting schemes, the objective is to provide a better framework for elucidation of gene-level interactions. This improves the value of the information obtained and will, in turn, permit refined parameterisation of the main model.

The second model focuses on epigenetic mechanisms. Research on Epigenetics is very active, but quantitative informative is still very sparse. Due to the complexity of the mechanisms involved and their interactions, model development is essential to linking biological and medical expertise. In that sense, the situation is similar to that of early HIV research, which focused on phenomenological explanations. Current limitations in available epigenetic information implies that a level of refinement similar to that of the agent-based approach used for the main layer is not a realistic target. What is needed first is, therefore, a *proof of concept* for such models, and our objective is to provide this, in the context of infection-induced epigenetic changes.

[3]Genes have specific functions. Even though each cell contains the whole genetic material, it only uses a fraction of this. The others are *silenced*. These complex dynamics are time dependent and cell-type dependent, and are referred to as gene expression.

1.4 Organisation of the thesis

The remainder of this Thesis is arranged as follows. Chapter 2 introduces concepts of immunity and immune response, and details the cells and organs involved. This Chapter also covers mechanisms linked with HIV infection, from virus structure to overall disease progression. This is intended to provide the reader with sufficient background to appreciate the motivations and challenges of the proposed model and the levels or layers which are required in its construction. Finally, this Chapter also provides a brief overview of existing immune models. This permits identification of limitations that need to be addressed through any new modelling approach.

Chapter 3 discusses the agent-based paradigm, associated challenges and relevant examples. It then introduces the approach taken, in particular detailing structure, behaviour and interactions of all agent types together with simulation procedures.

Chapter 4 focuses on the structures implemented to account for lymph nodes and the *lymph network*. Further background on phenomena involved is also provided, as are details on implementation of cell mobility.

Chapter 5 presents parallelisation efforts. Several strategies are tested and optimised, and scale-up is assessed through development of a performance simulator.

Chapter 6 details results obtained and evaluates success in addressing the limitations identified for existing models. It introduces a further model extension, the inclusion of localised effects in the gastrointestinal tract, and presents some initial results for this.

Further layers are added to the modelling approach, and consider gene expression as an underlying cause for phenotypical-level immune response. In particular, Chapter 7 focuses on epigenetic changes and their influence on gene expression. A brief background to this new research field is provided to introduce motivations and challenges. A model of infection-induced epigenetic perturbation is presented and validated.[4]

Finally, Chapter 8 summarises of key developments and results, and outlines future work.

[4]In Appendix A, we focus on microarrays and their use in analysing gene expression. In the interests of improving the quality of information feeding the central model layer, a novel weighting scheme for microarray analysis is proposed and evaluated, and a parallel genetic algorithm for biclustering is implemented and tested.

Chapter 2

Background and related work

2.1 Background on the immune system

2.1.1 Self and non-self: introducing immunity

Etymologically, immunity comes from Latin *immunitas*, from *immunis*: exempt from, (*in*), performing services, (*minus*)[1]. It was first used in the medical sense of "protection from disease" in the late 1870s (Harper, 2001).

In this particular context, immunity can be defined as a function of all mechanisms which permit the body to recognise entities belonging to its system, (which consequently it tolerates), and those that do not, (which it fights). There are many layers of complexity and response in the system, involving a number of entities and interactions. This poses a clear challenge to computer-based modelling, (as outlined e.g. in Forrest and Hofmeyr (2001)). In this Section, we introduce the cells and organs involved in the immune system.

2.1.2 Organs and cells of the immune system

Central lymphoid organs are the birth place of the immune system. There is where *lymphopoiesis*[2] takes place. Lymphoblasts are formed in the bone marrow, by differentiation,

[1] *minus*, "service performed for the community, duty, work".

[2] Lymphopoiesis: generation of lymphocytes. Details of cells and precursors can be found in the glossary.

from precursor hemocytoblasts. These are immature cells, from which prolymphocytes, direct precursors of lymphocytes, are derived. The last development step, from prolymphocyte to lymphocyte, can take place in two different locations, which will decide the final role of the cells: the prolymphocytes maturing in the bone marrow itself become B lymphocytes, while those maturing in the thymus become T lymphocytes (Potmesil and Goldfeder, 1973).

Once matured, lymphocytes are released into the blood circulation and migrate to peripheral lymphoid organs and tissues. These, primarily, are lymph nodes, but include also spleen and MALT, (Mucosa-associated lymphoid tissue), (Wiedle et al., 2001). Their associated network allows interactions between immune cells. While involved in cell renewal, (through cell divisions initiated after antigen recognition during immune responses), the main function of lymph nodes is to amplify immune responses.

At the cell-level, four populations are particularly important:

- T lymphocytes are divided in two families. CD4 T cells, or helper T cells (Th), coordinate the immune response. Two sub-types, Th1 and Th2, are in charge of the cell-mediated and antibody-mediated responses, respectively. CD8 T cells, or cytotoxic T cells (Tc) are "effector cells" of the cell-mediated response, (detailed below).

- B lymphocytes are effector cells of the antibody-mediated response. Their response can be initiated by activated CD4 cells, (often referred to as $CD4^+$ cells), but, contrary to CD8 cells, they can also directly recognise antigens. B lymphocytes display these antigens at their surface, making them antigen presenting cells.

- Natural killer, (NK), cells can have a direct cytotoxic action on abnormal cells, e.g. cancer cells or virus-infected cells. They form the major component of the *innate* immune system.

- Antigen-presenting cells, (APC), are all cells that can present antigens at their surface and which can, therefore, activate CD4 cells. These include B lymphocytes, but also

dendritic cells, macrophages[3], endothelium cells, (from the inner surface of blood vessels), and epithelium cells, (from cavities and surfaces of structures throughout the body, e.g. lungs, gastrointestinal tract, reproductive and urinary tracts).

Cells and associated functions can also be found in the glossary, for later reference.

2.1.3 Types of immune response

2.1.3.1 Innate immune response

When a foreign element is identified as a threat, it can be dealt with in two different ways: i.e. immune response can be *specific* or *non-specific*. A non-specific, or *innate*, response is based upon recognition of the pattern of the microbial surface components of the pathogens[4], rather than by a specific antigenic sequence (Levy, 1990). Innate response *does not confer* long-lasting immunity to the host, i.e. there is *no memory* of previous responses.

Major functions of the vertebrate innate immune system include:

- Recruiting immune cells to infection sites. This is achieved using *cytokines*[5].
- Identifying bacteria, activating cells and enhancing clearance of antibody complexes, through *complement cascade* (Barnes and Weiss, 2003; Sacks et al., 2003).
- Initiating the *adaptive* immune system, through antigen presentation to cells regulating the immune response (Martin and Carrington, 2005).

2.1.3.2 Antigen-specific immune response

In contrast to the innate response, the specific, or *adaptive*, immune response is based on the accurate recognition of foreign non-self antigens. As seen above, recognition involves

[3]Macrophages are a type of white blood cell that ingests foreign material. In that sense, they are involved in the non-specific immune response. They are also involved in the specific immune response: they carry the antigen on their surface and present it to a T cells.

[4]A pathogen is a biological agent which causes disease or illness to its host.

[5]Cytokines are a category of diverse signalling proteins and glycoproteins, which are essential to cellular communication. Other categories of signalling proteins include hormones and neurotransmitters.

innate response cells as intermediaries, and a weakened innate response may, therefore, lead to a delay in initiation of antigen-specific response. This is, for instance, observed for neonates[6], leading to higher susceptibility to pulmonary bacterial and viral infections (Garvy, 2003).

Antigen-specific response has two arms, namely *cell-mediated* and *antibody-mediated* responses. The latter, also known as the *humoral* response, features B lymphocytes as effector cells, and mainly targets bacterial attacks. Humoral response is characterised by production, by these cells, neutralizing antibodies, following activation by CD4+ T helper cells through release of *interleukin* IL-4 (Howard and Paul, 1982).

Cell-mediated response is targeted more specifically at viral attacks and takes place in lymph nodes. It is, therefore, the main focus of this work. Cell-mediated response involves, (Kaufmann, 1999; Pathak and Palan, 2005):

- Activation of antigen-specific cytotoxic, (CD8), T lymphocytes, which induce apoptosis in body cells displaying epitopes[7] of foreign antigen on their surface, (e.g. virus-infected cells, or cancer cells displaying tumor antigens);
- Activation of macrophages and natural killer cells, which then destroy intracellular pathogens;
- Secretion of a variety of cytokines that enhance function of other cells involved in adaptive immune responses and innate immune responses. This is the role of CD4+ T helper cells.

Prevalent in the fight against HIV are CD8+ T cells. This particular response involves three steps. Antigen Presenting Cells (APC) acquire foreign biological entities and start presenting these antigens at their surface. These encounter CD4 lymphocytes which will self-activate and multiply, *if designed to recognise* the given antigen. These, in turn, activate specific CD8 cells, which will multiply and then target infected cells.

[6] A human infant less than four weeks old.

[7] An epitope is a protein site which is recognised by the immune system, (specifically by antibodies, B cells, or T cells). For simplicity, an epitope can be considered as the 3D surface features of a molecule. These "shapes" fit precisely and thus bind to specific antibodies. Epitopes are also known as *antigenic determinants*.

2.2 Mechanisms linked with HIV infection

2.2.1 HIV is a retrovirus

HIV belongs to a very particular viral family, *Retroviridae*. Traditionally, in biological processes, DNA is transcripted as RNA, which is then used as an intermediary for protein production. This is known as the *central dogma* of molecular biology (Crick, 1970). The defining feature of these retroviruses is that they use the opposite process: they possess an RNA genome and replicate using a DNA intermediate, (called a *provirus*). This reverse transcription is performed using an enzyme[8], reverse transcriptase[9], and a second enzyme integrates the obtained DNA into the host's genome[10]. These transfers of biological information are shown in Figure 2.1.

Original classification of retroviruses is based on their pathogenicity and include oncoviruses[11], spumaviruses[12] and lentiviruses[13], which are characterised by the slow progression of the infections they induce. HIV belongs to this sub-family.

2.2.2 Structure and genome

As with other retroviruses, HIV has two copies of its RNA genome. This genome contains nine genes, with three of them, (*gag*, *pol* and *env*), characteristic of all retroviruses, and containing information needed to make the structural proteins for new virus particles (Lever, 2005). The other genes e.g. control transcription activation (*tat*), counteract the antiretroviral defenses of the host cell (*vif*), or enhance virus release (*vpu*) (Sierra et al.,

[8] Enzymes are molecules which increase the rates of chemical reactions. This is referred to as a catalytic action.

[9] A reverse transcriptase is an enzyme which transcribes single-stranded RNA into double-stranded DNA. This process is the *reverse* of the normal transcription, which corresponds to the synthesis of RNA from DNA. These enzymes are also known as RNA-dependent DNA polymerases.

[10] The genome of any living organism is its whole hereditary information. It is encoded in the DNA or, for some viruses such as HIV, the RNA.

[11] Oncoviruses are the largest sub-family of retroviruses. They can induce several types of tumours, e.g carcinoma, lymphoma and leukemia. They have been isolated in humans as early as 1980s, (see e.g. Rho et al. (1981)).

[12] Spumaviruses, also known as foamy viruses, are non-pathogens. They are mainly prevalent in non-human primates, and were first described in the early 1950s. They are easily isolated, thanks to the characteristic foam-like effect they induce (Delelis et al., 2004).

[13] Lentiviruses are cytopathogens and are responsible for slow-progression infections, hence their name.

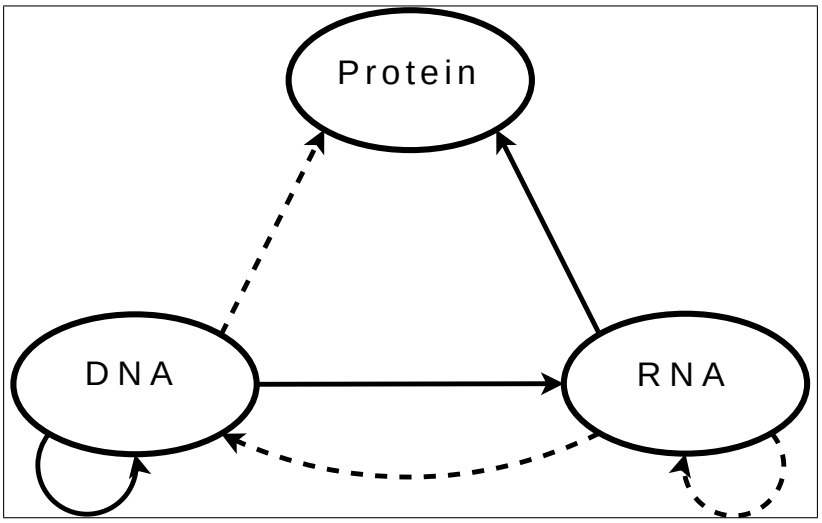


Figure 2.1: Information flow in biological systems.
Plain lines represent general transfers, (occuring normally in most cells), and dotted lines represent special transfers (known to occur, but only under specific conditions, e.g. some viruses or during *in vitro* experiments). For retroviruses, transfer of information from RNA to DNA is occurring.

2005). This genome is enclosed by a conical capsid, which is surrounded by a matrix, in turn surrounded by a viral envelope.

An important glycoprotein complex is embedded into this viral envelope. It is made of six glycoprotein (gp) molecules, (three gp120 and three gp41), and enables virus attachment and fusion with target cells (Chan et al., 1997). This complex clearly appears in Figure 2.2, which shows the overall structure of the virus.

2.2.3 From cell infection to release of new virions

A complex sequence of mechanisms are involved, from the infection of an immune to the release of new virions. This is summarised in Figure 2.3, (p.14). These mechanisms have been studied in detail, (see e.g. Wyatt and Sodroski (1998)), so we focus, in what follows, only on essential points.

The first step in infection of macrophages or CD4+ T cells is high-affinity attachment of the CD4 binding domains of their surface glycoprotein gp120 to CD4 receptors. This is fol-

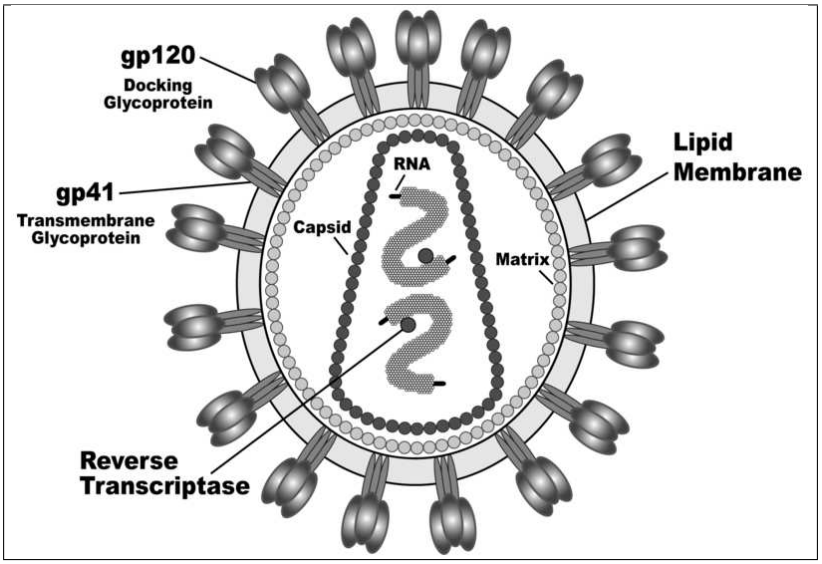


Figure 2.2: Structure of HIV virus (adapted from a public-domain image from the National Institute of Health, USA)

lowed by interactions between chemokine[14] binding domains of gp120 and the chemokine receptors of the target cell. It must be noted that, even though the prime target for HIV infection is CD4 cells, HIV can also infect dendritic cells using other specific receptors such as DC-SIGN (Wyatt and Sodroski, 2001). This alternative route is believed to play an important role in early stage of infection (Pohlmann et al., 2001). Even though the binding process is different, subsequent steps in cell infection are similar.

The viral envelope and cell membrane are fused. This allows release of viral capsid into the cell. Viral RNA and essential enzymes, e.g. reverse transcriptase, are then transferred into the cell, and RNA is transcribed into double-strand DNA. Finally, This DNA material is then integrated into the cell genome.

This reverse transcription is a fast process, (generation of 10^9 to 10^{10} virions every day), but this high rate leads to many *transcription errors*, i.e. *mutations*, (approximately 3×10^{-5} per nucleotide base per cycle of replication), (Robertson et al., 1995). New variants are created every day in any individual patient.

[14]Chemokines are a family of small cytokines. They induce directed *chemotaxis*, (innate movement), in nearby cells, hence their name.

It is evident, therefore, that HIV has a very high genetic variability. It is categorised by two species, HIV-1 and HIV-2, where the latter is both less virulent and less transmittable (Reeves and Doms, 2002), but the former is responsible for much of the global infection and attracts most research effort. In the remainder of this Thesis, unless stated otherwise, use of "HIV" refers to HIV-1.

Newly integrated viral DNA remains dormant. Active production of new virions only starts in the presence of certain transcription factors. Ironically, these factors are upregulated when cells are activated, (Nabel and Baltimore, 1987), so that virus production is increased, and the infection progress catalysed, by the immune response resisting it.

Viral RNA is produced in small sections, which are then assembled (Wu and Marsh, 2003). This leads to the production of large polyproteins, which are cleaved into smaller structural proteins, which assemble near the cell inner membrane and form a bud, subsequently released from the cell (Cimarelli and Darlix, 2002). Maturation, (i.e. obtention of functional proteins and enzymes), can take place either in the forming bud, or after the virion is released from the cell, (see e.g. Ohagen et al. (1997)). Once mature, a virion is then able to infect a new cell.

2.2.4 Progression of infection

Macroscopic evolution of the disease is divided into three phases, starting with acute HIV infection. This is characterised by rapid viral replication and high presence of virus in the peripheral blood. This results in massive activation of CD8+ T cells, which target and destroyed HIV-infected cells. The development of HIV-specific immune responses, and "consumption" of available targets, by the virus, result, within a few weeks, in peak and decline of the viral population (Derdeyn and Silvestri, 2005; and references therein). The virus is not eliminated though, and has been widely disseminated, and seeded in lymphoid organs, (because of high target cell concentration in those regions).

Once most original strains are eradicated, the importance of the *mutation rate* becomes critical. It allows appearance of new strains, which have not been detected by the organ-

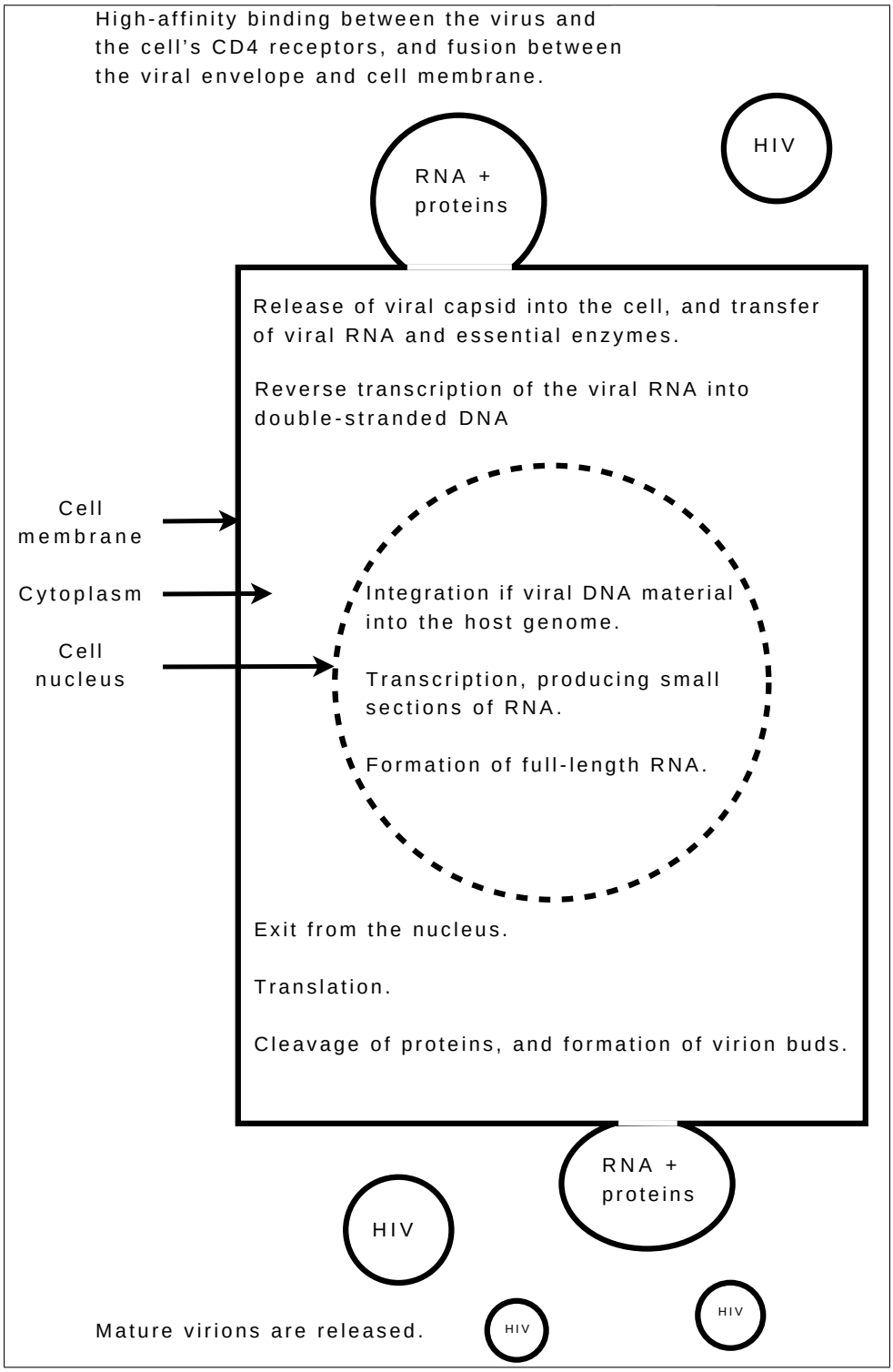


Figure 2.3: Cycle of HIV infection

ism yet, and can therefore develop freely. As soon as a strain becomes too intrusive, its detection probability increases and it is eradicated, but in the meantime, new strains have again emerged. During this second phase, there are no visible symptoms. This is known, (somewhat erroneously), as the *latency period*, and can last several years. The immune system is heavily loaded, and a state of chronic, generalized immune activation develops. This, added to the fact that destruction of a strain implies destruction of all cells it infected, leads to T-cell depletion (Hazenberg et al., 2000).

Once CD4+ T cell count declines below a critical level, (typically defined to be between 200 and 400 cells/mm^3, see e.g. MacDonell et al. (1990)), the cell-mediated immune response can no longer be initiated, and a variety of opportunistic diseases occur, signatures of full-blown AIDS, (acquired immunodeficiency syndrome), ultimately leading to the death of the patient.

2.3 Mathematical and computational models: historical

2.3.1 A complex biological system

The immune system is both a *complex* and *adaptive* system. Not only does it involve various cells and organs, and interactions between these, but its behaviour can also evolve over time, changing and learning from experience, through memory of past immune responses. Complex adaptive systems are encountered everywhere, including other biological systems such as RNA folding (Ndifon, 2005), but also in social systems (Janssen and Ostrom, 2006), ecosystems (Horwitz and Wilcox, 2005), manufacturing (McCarthy and Tan, 2000), and financial analysis (Sharkasi et al., 2006; Thurner and Biely, 2007), among may others. Key principles of complex adaptive systems are *emergence* and *self-organisation*.

Emergence refers to patterns of system evolution which arise from an abundance of simple, low-level, interactions (Crutchfield, 1994). (In the context of the immune system, this is particularly relevant, as the response is obtained from the multiplicity of cell interactions throughout the body.)

Self-organisation refers to increased complexity obtained without intervention from an outside source (Goertzel, 1992). (Again, this defines both the infection mechanisms of HIV and the immune response to these.)

Given these properties, complex adaptive systems are often difficult to describe fully. In particular, defining the contribution and importance of low-level unsupervised interactions to the overall evolution process is far from trivial. Consequently, several models are proposed in what follows. Concepts and approaches are detailed, with a particular focus on immune models.

2.3.2 Bottom-up vs. top-down design

2.3.2.1 Concepts

Two categories of complex system modelling are discussed, *top-down* and *bottom-up* designs (Bohringer and Rutherford, 2008).

The main concept of a top-down design is to break down a system into several components, which are expected to be easier to manipulate and understand. The overall system is formulated and specified, but without going into details of its parts. In an iterative process, each component is then defined in more detail and, if necessary, split into lower-level subsystems. This process, repeated until entire specification is obtained for base elements, involves use of *black boxes* which facilitate model development, but may also hinder model validation if they fail to elucidate elementary mechanisms of the system studied.

In a bottom-up approach, individual base components are detailed and designed, and then linked together. These form more complex systems, which are again linked, in an iterative process, and the top-level model increasingly emerges. This approach is, therefore, particularly suited to complex adaptive systems, which in their structure already show emergence and self-organisation.

The remainder of this Section considers several examples of these two approaches, grouped in three families: mathematical, shape-space, and agent-based models.

2.3.2.2 Mathematical models

In the context of HIV research, mathematical models were first introduced to study the epidemiological aspect of the infection, (i.e. spread in a human population).

This early focus was motivated by the need to understand dynamics of the infection and population threat, but also by a lack of detailed biological information, which ruled out models of pathogenesis. Although data even on spread of HIV were sparse, models were developed, which focused on specific "at risk" groups, e.g. early work by Anderson (1988). More accurate medical information and improved computing resources have now led to considerably more advanced epidemiological models, e.g. Naresh et al. (2006).

In the context of this research, the *mathematics of pathogenesis* are directly relevant. Models using differential equations, (DE), to reproduce variations of cell counts and viral loads appeared in the late 1990s, (see e.g. Perelson and Nelson (1999)) and are typical of top-down designs. These have been refined for each count variation by detailing different DEs to account for viral production, and drug therapies such as RT inhibitors or protease inhibitors. An equation describing the rate of change of infected $CD4^+$ T cells (T^*), for instance, would have the following structure:

$$\frac{dT^*(t)}{dt} = (1 - \epsilon_{rt})kT_0V_i(t) - \delta T^*(t) \tag{2.1}$$

This corresponds to the virus infecting $CD4^+$ cells at a constant rate k, less the currently infected $CD4^+$ cells, which die at constant rate δ.

These models, however, can not currently cover the whole infection progression, and each focuses on either long-term, mid-term, or short-term variations.

2.3.2.3 Shape-space models

Immunological models based on the shape-space paradigm were first introduced as a means to account for dynamics of antibody-antigen bindings (Perelson and Oster, 1979).

The main concept of this bottom-up approach is to represent each clonotype by N integer

parameters and, therefore, to consider clonotypes as points in an N-dimensional Euclidiean space. In that *shape space*, (space of clonotypes), two cells sharing the same clonotype are located on the same point. Each cytotoxic lymphocyte clonotype c is surrounded by a sphere of radius r. To any antigen a within this sphere is applied a clearance pressure inversely proportional to the distance between a and c in the shape space.

Technical considerations such as adequate values for N are still subject to intense discussion, which will not be detailed here, in order to avoid a digression which does not focus primarly on models of interest. Several papers include models with $N \in [1, 5]$, (see e.g. Papa and Tsallis (1996)), while on rare occasions others impose $N \geq 20$, (see e.g. Carneiro and Stewart (1994)), but no clear consensus emerges.

The most recent and interesting examples take the shape-space paradigm further. These include considerations of real space, and formation of hybrid models (Burns and Ruskin, 2004; Ruskin and Burns, 2005; 2006). The focus is on emergent principles of CD8 cell clonotype repertoire and its distribution and differentiation, with emphasis on systemic self-organisation, (for which the shape space paradigm is particularly suited). Here, clonotypes and viral epitopes are represented as nodes in a two-dimensional *network space*, and edges between nodes again model the affinity and clearance pressure applied to the APC which bears the target epitope. *Hybridisation* of shape space is obtained through use of a stochastic model of the lymph system as stimulus to the *network shape space model*. Emergent topology obtained from this model resulted in introduction of a theoretical network architecture for immune system shape space. It includes α and β nodes: the latter correspond to CD8 cells that act only against the antigen, which stimulated its activation, while the former represent those which also effect clearance pressure on subsequent APCs. The argument outlined suggested that disruption (or suppression) of α nodes results in a signicantly degraded pathogen clearance, compared to β node disruption. This was proposed as a possible cause of individual variations in the latency period.

2.3.2.4 Cellular Automata and Agent-based models

The Cellular Automata, (CA), paradigm is another popular example of bottom-up model design. Even though the concept was introduced in the 1940s, (through the work of Ulam and von Neumann), the paradigm gained large popularity only in the 1970s, with the introduction of a two-state, two-dimensional cellular automaton, Conway's "*Game of Life*" (Gardner, 1970). On a 2D grid, cells have eight neighbours, and two possible states, *live* or dead. At each time step, cells are updating using simple defined rules, e.g. a live cell with fewer than two live neighbours dies, while a live cell with more than three live neighbours dies.

The popularity of this interpretation of the paradigm can be attributed to its obvious analogy with living systems, (e.g. rules which embody death by overcrowding or competition), but also to this CA being a perfect and simple illustration of concepts of emergence and self-organisation. Pattern evolution in this CA is well documented, with example entities such as *blocks*, *gliders* and *pulsars* (Berlekamp et al., 2004).

More realistically, when including more than two possible states, Cellular Automata provide a powerful modelling paradigm. Since the early efforts of e.g. Celada and Seiden (1992); Seiden and Celada (1992), CA have been widely used to investigate immune events, and several models in particular provide useful insights into some aspects of the immune response to HIV infection.

CA-based immune models outlined the importance of viral mutation on dynamics of immune cell population (Mannion et al., 2000; Pandey et al., 2000; Mannion et al., 2002; Ruskin et al., 2002). A threshold value was identified, under which steady-state density of immune cells is larger than that of the virus, (i.e. dominant to deficient phase transition). These authors also provided one of the rare attempts to account for cell mobility and, subsequently, variable viral load.

Another CA model focused on latency period and treatment solutions (Benyoussef et al., 2003). Combining a mean field approximation method and CA simulations, these authors reproduced the three-phase evolution of HIV infection and identified a threshold for treat-

ment, (a combination of protease inhibitors and RT inhibitors), above which virus load decreases over time. They also indicated that such treatment would need to continue for years even if viral load falls under detectable limits, (this is, we think, a direct consequence of blood samples not being an accurate reflection of disease progression within lymph nodes). The agent-based paradigm can be considered as an extension of the CA approach, but previous agent-based models of immune events have been limited in implementation. A brief review follows.

An early example of such models, (Bernaschi and Castiglione, 2001; Castiglione et al., 2004; Baldazzi et al., 2006), is, in strict terms, closer to a CA than to an agent-based model, but the authors themselves use both terms, and it is more important to focus, here, on implemented features rather than terminology. The authors main objective is development of a simulator including features introduced in the Celada-Seiden model, (IMMSIM), but with various refinements, particularly in terms of performance and fidelity to the real immune system. Implemented entities include CD4 and CD8 T cells, B cells, macrophages and dendritic cells. These interact, based on location in a 2D or 3D lattice, and according to an affinity function that depends on values of bit strings representing their respective *binding sites*, (e.g. lymphocyte receptors, or class I and II major histocompatibility complexes[15]). Interactions lead to changes of internal state. The time step for these simulations is eight hours. This approach successfully reproduced the three-phase disease progression.

Another successful attempt at reproducing typical HIV progression was recently proposed (Zhang et al., 2005). Here, types of agents are limited to T cells and HIV virions, with the objectives of simulating large populations, (hence the term "massively multi-agent"), and improving detail on aspects such as immune memory and HIV sequence representation. The former objective is achieved through introduction of a global memory repertory, which is empty when simulation starts and which subsequently stores all HIV genomes that have

[15]There are two primary classes of major histocompatibility complex molecules, MHC class I and MHC class II. MHC class I molecules are found on almost every nucleated cell of the body. They present antigen fragments to cytotoxic T-cells and bind to the CD8 receptor on these cells. MHC class II molecules are found only on antigen-presenting cells. They present antigen fragments to T-helper cells by binding to the CD4 receptor on these cells. In humans, these MHC molecules are sometimes referred to as human leukocyte antigen (HLA) molecules.

been recognized and were targeted by immune responses. The latter objective is achieved by introduction of finite-size binary arrays, which each represent a viral strain, a measure of similarity between these arrays, (based on Hamming distance), and a "recognition probability" as a function of this similarity.

2.4 Limitations and proposed focus

2.4.1 General considerations

Models of HIV progression have been developed for the last twenty years, with some successes reported, as detailed above. There are, however, a number of limitations that need be addressed in order to improve realism with respect to the biological system modelled.

A major limitation of top-down models, especially mathematical ones, is that ability to deal with subsystems small enough to be informative on the interactions involved is crucial in the context of HIV progression. Immune response typically is bottom-up, and these models may not, therefore, be the most suited to describe it.

The shape-space paradigm is very elegant, and hybrid models based on this provide new insight into self-organisation of the immune response. It is, however, sometimes difficult to explain observed emergent features in biological terms (Burns, 2005). This may be attributed to the fact that such models, even though bottom-up, can rarely achieve sufficient refinement of the individual base components. Extensions proposed to current models may improve this situation (Burns, 2005).

CA and agent-based design seem well-suited to the nature of the biological system being considered and recent biomedical studies, involving direct tracking of HIV viral genotypes in local microenvironments, provide further validation for these approaches, (e.g. infected cells releasing virions will only induce infection of local targets (Cheynier et al., 1994)). It seems evident that local interactions and cell densities are more important to disease progression and experience than overall cell counts in the body (Grossman et al., 1998). As a bottom-up approach, the agent-based paradigm offers the best prospects for detailed local

solutions and has been adopted for the work of this Thesis.

2.4.2 Advancing the agent-based approach: limitations and improvements

Balance between diversity and population size

Inherent to any agent-based approach is choice of agent types, where several permit improved tuning of the model to the observed system. It is particularly relevant in the context of immune responses, as several lineages of cells are involved. However, there is a balance to be achieved between the range of agent types and size of their populations, due to the computational cost involved. This is important in the immune context, as overall behaviour emerges from a very large number of interactions, (involving many cells). It is not clear that sufficiently large cell populations have hitherto been modelled. The theoretical limit in Bernaschi and Castiglione (2001) is not explicitly specified, but appears to be of the order of two million, and it is debatable whether this is enough to account for HIV progression and complex evolution through the whole body.

Lymph nodes modelling

Another essential aspect is that of spatial location of immune response elements. Most of the immune response to HIV is taking place in the lymph nodes, as opposed to within circulating blood, but no clear modelling approach has been explicitly described for this. There are indications, in Baldazzi et al. (2006), that overall structure is recognised as important, but implementation remains vague and no report of validation tests could be found at the time of writing. Any novel agent-based approach needs to explicitly account for lymph nodes and, specifically, the *lymph network* in order to gain insight on the importance of cell mobility[16]. The proposed model is, therefore, organised as shown in Figure 2.4.

[16]Such compartmentalised approaches were also considered in the context of HIV epidemics, (see e.g. Ball et al. (1997) who considered transmission through a network whose nodes are families, which are assigned higher individual-to-individual transmission probabilities). The main difference, however, is that distant infection from a lymph node to another is not possible: physical contact between the cells is needed, and the force of the infection can not be directed. Their results, therefore, are not directly relevant to our study.

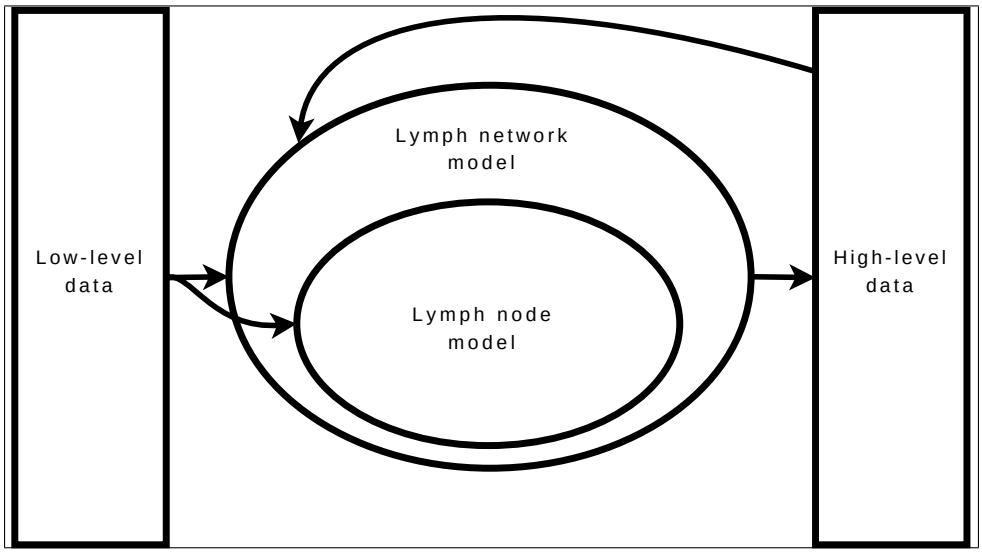


Figure 2.4: Overall structure of the lymph network model
The model is based on an explicit lymph node model. This key unit is used to locate the agents and implement their interactions. Low-level biomedical information, (e.g. microscopic interactions detailed above), are used to design both model levels. High-level information, (e.g. structure of the lymph chains), is also used for the lymph network model. The overall model produces high-level data, such as cell counts.

Temporal granularity

The proposed structure in Baldazzi et al. (2006), although non-specific, could provide a solution basis, as it includes vessels between lymph nodes, but the long eight-hour time step of the associated simulations seems unrealistic, especially in the context of a bottom-up design and need for detailed modelling of cell-level interactions. The temporal granularity is not refined enough to account for cells passing through these vessels, (which occurs within minutes), so that the context for inclusion does not add to realism of the model. Small time steps are required, even though this adds to the overhead on code in order to increase computational efficiency.

These two conditions, (explicit structure for the lymph network, and high granularity), permit accurate implementation of cell mobility, which is an essential aspect of immune response, as activation involves not only affinity between epitopes, but also physical contact.

Immune memory, and antigen recognition

Memory of previous responses is also important, (see Zhang et al. (2005)), but the implementation here is slightly counter-intuitive, and features a centralised control. Memory at cell-level is potentially more useful in light of the role of localised interactions.

A common feature of the most advanced models is incorporation of viral mutation. This is core to viral model success but consideration of variable properties, (e.g. to account for less stable strains, or for those with higher probability of successful infection), is also important for sophisticated model development.

Finally, current implementations for antigen recognition are based on lock-and-key or on naive distances, and response is binary: there is complete recognition, or no recognition at all. This contradicts recent immunological understanding (Brehm et al., 2002), and requires explicit improvement in any newly-proposed model.

Localised effects

Including all these new features requires extensive code optimization, and parallel implementation, and the costs are, additionally, a longer development process and additional tests. The gains, however, are significant. The objective is a model which permits large-scale simulations and inclusion of layered, localised, effects, such as those found recently for the gastro-intestinal tract, (see Chapter 6), and which accounts for multiple facets of the biological immune response.

2.5 Chapter summary

In this Chapter, an overview of essential entities and mechanisms, associated with immune responses and HIV disease progression, was presented, together with a brief review of previous efforts to model this complex adaptive system.

In our view, the agent-based paradigm is most suited to describe the way in which complex combination of cell-level interactions create individual variations in length of the latency

period, but challenges linked with this approach have been highlighted. They include modelling choices at all levels of implementation, from cell level, (i.e. decentralised immune memory, refined antigen recognition) to organ level, (i.e. explicit lymph node implementation, inclusion of localised effects), and simulation strategies, (balance between agent diversity and population sizes, refined temporal granularity).

Any new model will need to address these issues in order to gain further insight into the biological system. The following Chapter describes how the local model structure can be part of the solution to gaining key insight into the overall biological system behaviour.

Chapter 3

Basis of the agent-based structure

3.1 Concept of agent-based models

3.1.1 Agents and agent-based formalism

An agent-based model is a model in which the key abstraction elements are *agents* with both spatial and temporal positioning. The generally-accepted properties for an intelligent agent, there being no unique definition, are given by Wooldridge and Jennings (1995):

- *Autonomy*: it can act without any intervention and has some control over its actions and its internal state.
- *Social behaviour*: it can interact with other agents through a specific language.
- *Reactivity*: it can scan part of its environment and change its behaviour to take advantage of it.
- *Proactivity*: it not only reacts to its environment but also acts and takes initiatives, to satisfy identified goals.

It is theoretically possible, though counter-intuitive, to design an agent-based approach and implement it without explicit references to agent-like entities in the code. In practice, efficient realisation of the agent-based paradigm requires both.

Advantages of the approach include modelling efficiency, robustness, interoperability between existing systems, and reasonably intuitive solving of problems for which data, expertise and control are distributed Jennings et al. (1998). The approach is this particularly useful in the context of Natural Sciences, as it permits reciprocity between agents and biological entities as well as between interactions of the real system and exchanges between agents.

3.1.2 Agents, environment: coordination and challenges

Communication modes

Each agent has limited access only to information about the environment it evolves within, and agent sociability, therefore, is a crucial aspect of abstraction. Agents can communicate in several modes, as detailed in Chaib-Draa and Dignum (2002) and summarised, for convenience, in Table 3.1. Not all systems involve the complete set of communication modes, but all modes used in a given system must be identified and explicitly managed.

Communication mode	Definition
Representative	Providing information on a state
Directive	Commanding or asking the recipient to perform an action
Commissive	Leading the agent to commit itself to performing an action
Expressive	Defining a "psychological" state
Declarative	Adding something to the environment
Permissive	Granting permission for an action
Prohibitive	Opposing an action

Table 3.1: Agents can communicate in several modes

Cooperation vs. competition

Excluding very rare exceptions, an agent always shares its environment with other agents. It is, therefore, necessary to coordinate all actions of the multitude of agents. Of course, coordination does not imply cooperation. Cooperation is unnecessary: an agent may oppose another in the sense that advantage may be taken to coordinate actions in response to

a competitor's decision. Reciprocity is also not implied as a decision, affecting e.g. movement to a proximate location, need not be influenced by an agent already in that location. In any coordination strategy, the size of the agent population is fundamental and, if every agent can mutually interact, the number of interaction pairs increases quadratically with the population size. If interaction can occur between *several* agents, the coordination overhead increases exponentially and soon challenges available computing facilities (Durfee, 2001). Even on recent, large-scale computing resource, developing a coordination strategy is therefore both essential and non-trivial, and avoidance of conflicts and blocks is often as much as can be managed. As already noted, the main drawback of the agent-based approach is the fact that it is resource-greedy; for this reason, parallelism is desirable, (Chapter 5).

Other challenges

Several challenges exist, (see e.g. Bond and Gasser (1988); Franklin and Graesser (1997); Iglesias et al. (1997); Sycara (1998); Chaib-Draa et al. (2001)). In particular, formulation, description and decomposition of problems are non-trivial. This is common to most bottom-up approaches. Analysis is, similarly, not evident, and requires, where possible, the elimination from the global system of chaotic or oscillatory behaviour.

Even on parallel implementations, resource allocation is challenging, especially when dealing with a large agent population. In particular, when some resources are limited, sensible allocation of these is required. For instance, a balance between a local treatment by a single agent and a solution involving several agents, (and, therefore, communication), must be achieved.

Finally, agent sociability is also non-trivial in most cases. Each agent must be able to represent its actions and reflect on them, but also to communicate on these with other agents. This implies possible conflictual intents and divergent points of view, which must be managed during the coordination of agents. Similarly, each local decision an agent takes has global implications. Damaging interactions must be identified and deleted.

3.1.3 Examples

Agent-based models implementing several agents are referred to as *multi-agent systems*. These systems provide a generic framework for model development and have been widely applied. A brief review follows.

One of the earliest, and most productive applications was air traffic scheduling. In the model proposed, (Cammarata et al., 1983), each agent represents a flight and the objective of each agent is to build a flight plan which maintains a minimum security distance with neighbouring planes, but also satisfies a number of additional constraints such as minimizing fuel consumption. In cases of conflict, (e.g. of flight plans), one selected agent acts as a central negotiator to determine new options for each agent. This agent is chosen based on a set criteria such as best informed agent, or agent with smallest number of constraints. A related problem, dealing with optimum service to the public under constraints, (e.g. maintenance and crew requirements), was solved using an agent-based approach (Langerman and Ehlers, 1997), with agents representing origin and destination airports, and interacting for resources.

As an illustration of application in the public sector, GUARDIAN was developed to solve structural issues in the context of an intensive care unit (Hayes-Roth et al., 1989). Here, unit structure is based on information exchange between experts in their own field who collaborate with the common objective of deliver efficient patient care. GUARDIAN implementation is based on three types of agents; perception/action agents; organisation/decision taking agents; high-level system control, (performed by a single agent).

In DVMT (Distributed Vehicle Monitoring Task), the paradigm is applied to a radically different field: agents are scattered over geographic areas, which they monitor (Durfee, 1998). Over regular time steps, each agent detects the characteristic noise emitted by vehicles. Based on this, it draws a description of vehicle movements throughout its allocated area. Communication between agents is then used to increase reliability and prevent redundancy of information in areas monitored by multiple agents. Communication is implemented using a *blackboard*.

Agent-based programming can also be used to control robots (Mizoguchi et al., 1999). Actions require high refinement of communication, negotiation, collaboration, decision-taking and execution, as robots interact both with each others and with humans.

Finally, while these examples are developed *ab initio*, there are several development environments which can be used for agent-based models, such as Swarm (Minar et al., 1996), Cougaar (Helsinger et al., 2004) or JAMES (Uhrmacher et al., 2000). The latter, for instance, is a Java-based framework aimed at modelling and simulation, and permits creation of mobile agents. These development solutions are not suitable in the context of this study, which demands a very large agent population, with extensive optimization requirements at every level of implementation.

From the few examples above, (the list is not intended to be exhaustive), it is evident that the agent-based paradigm is a versatile approach. In the next Section, details are given on applying this efficiently to immune modelling.

3.2 Model structure

3.2.1 Modelling approach

Object-oriented modelling techniques describe a system by identifying key object classes in an application domain and by specifying their behaviour and their relations with other classes. Essential details of a system are, therefore, traditionally reported using an *object model*[1], a *dynamic model*[2], and a *functional model*[3].

The proposed model of the immune system is implemented using *C++*, and could be described using models of all three types. In the context of agent-based systems, a more efficient specification technique operates on levels of abstraction (Kinny and Georgeff, 1996). At the external level, the system is divided into agents, which are modelled as complex

[1] An object model gathers all details on objects within the system, describes their structure, their relations and the operations they support.

[2] A dynamic model describes the states, transitions, events, actions, activities and interactions of the system structures, which characterise system behaviour.

[3] A functional model describes data flow during system activity, both within and between components.

objects characterised by their aim, their responsibilities, the services they provide, and their interactions with the environment. At an internal level, required elements for each agent type and structure are modelled.

For clarity and efficiency, the proposed model is detailed using this specification technique.

3.2.2 External structure

At the external level, two specifications are provided (Kinny and Georgeff, 1996). An *agent model* describes hierarchical relations between abstract and concrete classes, and identifies agent types involved in the system, their populations, and the dynamics of these. An *interaction model* describes responsibilities of an agent type, services it provides, associated interactions and control relations between agent types.

3.2.2.1 Agent model

Balance between agent diversity and agent population is essential, (Section 2.4.2). In order to simulate populations large enough to account for immune response throughout a body, reduced diversity is needed, at least in the first instance. As immune response to HIV is predominantly cell-mediated, it can be modelled using three types of agents, (based on agent-cell reciprocity): CD4, CD8 and APC. A fourth type of agent is needed for HIV virions. The agents evolve in lymph nodes modelled as 2D matrices in which each element is a 3D physical neighbourhood able to host several agents of all types. (For the lymph network environment, see Chapter 4.)

Each agent type corresponds to a specific class in the *C++* implementation, and inherits from a common, abstract base class. This class contains attributes and functions needed for management of ageing and of location within matrices. Being abstract, this class does not correspond to any agent during simulations. For immune (CD4, CD8, APC) and viral agents, the number in each matrix element is limited to 10 and 20, respectively, to ensure realistic neighbourhood size for interactions. An element can, for instance, contain 8 CD4 agents, 7 CD8, 5 APC and 17 virions.

At the start of a simulation, initial agent populations are specified in a parameterisation file. Agent populations then dynamically evolve, following rules summarised in Table 3.2.

Variations	Viral agents	CD4 agents	CD8 agents	APC agents
Increases	Production by infected CD4	Created in thymus, or produced by multiplicating agent	Created in thymus, or produced by multiplicating agent	New agent created
Decreases	Agent ingested by APC, or infecting CD4	Infected agent destroyed, or end of life	End of immune response, or end of life	Presenting agent destroyed, or end of life

Table 3.2: Agent population dynamics

3.2.2.2 Interaction models

The reciprocity between agents biological entities, (cells and virions), mirrors real-system interactions between the latter. These are summarised in Figure 3.1.

3.2.3 Internal structure

At the internal level, each agent class is described using three *specification models* of the agent-based paradigm. Following the structure proposed in Kinny and Georgeff (1996), the chosen approach is to use BDI models, (Belief-Desire-Intention), which describe information possessed by the agent, together with its objectives and potential behaviour. These "models", therefore, guarantee an exhaustive description of essential properties of the agent. The *belief model* describes the type of information the agent has access to, both in its environment and its internal state. In particular, several states can be specified as initial states. The *desire model* describes objectives of the agent, and events it can react to. Finally, the *intention model* describes the behaviour of the agent, i.e. ways in which the agent can fulfil its objectives or adapt to events it perceives.

These "models" will be used to describe each agent class: virions, CD4, CD8, and APC.

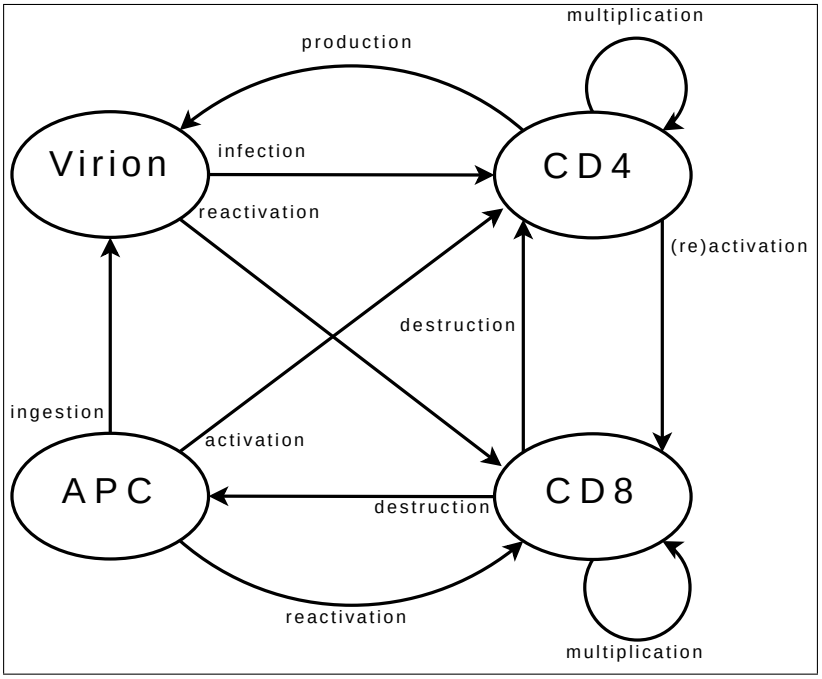


Figure 3.1: Agent interactions.

Viral agents can move from one neighbourhood to the next, (providing there is sufficient space for them, which is checked using a dedicated function). Upon arrival in a new neighbourhood, these agents can perform a single operation only: infection of a CD4 agent. First step is the selection of CD4 target. Possibility of infection is then assessed, (e.g. if CD4 agent is activated). Infection, if it takes place, is implemented as transfer of viral genome information into the CD4 agent and destruction of viral agent. **CD4 agents** incorporate mobility and can reach new neighbourhoods. As for viral agents, this includes checking, through a dedicated function, that space is available. Upon arrival and if already activated, the agent can activate a CD8 neighbour. Activation follows a process similar to that of infection, detailed above. A possible target is selected, assessed, (in terms of agents bearing "compatible" clonotypes), and then activated when this is possible. If a CD4 agent is infected, it may produce a new viral agent. In the early stages after its own activation, an agent also produces some additional CD4 agents, to enhance immune response. **CD8 agent** mobility is implemented similarly. In its new neigbourhood, the agent produces new CD8 agents, (if it is currently multiplicating), or targets infected CD4 and APC agents, (if it is activated). Some CD8 agents can enter a state representing *memory cells*. These agents, (with a greater life span and faster reactivation), interact with all agent types in their neighbourhood, in order to monitor known viral strains, and can be directly reactivated. Final agent type, **APC**, implements mobility and associated functions needed to query the environment. APC agents interact with viral agents, in the sense that they can ingest them to present viral strain information to other agents. They also interact with CD4 agents, which they can try to activate by presenting viral information. Success of activation is based on affinity between viral epitope and CD4 clonotype.

3.2.3.1 Belief-model specifications

Each agent has complete knowledge of its parameters and, therefore, of its internal state. Knowledge on its environment is, however, limited and temporary. Information on the environment is, indeed, limited to presence of targets for interaction in the neighbourhood. Agents have no memory of the evolution of their neighbourhood, as is the case with biological entities of the real system.

All internal parameters are coded as integers, (or lists of). Age is involved in the internal state of all agents. Other parameters are type-specific, and are given in Table 3.3.

Agent type	Parameter	Value stored	Initial value
Virion	Viral strain	Strain ID	Fixed value
CD4	Clonotype	Clonotype ID	Fixed value
	Activation	Responsible strain	0
	Multiplication	Expansion status	0
	Infection	Responsible strain	0
	Memory	Past activation status	Fixed value
CD8	Clonotype	Clonotype ID	Fixed value
	Activation	Responsible strain	0
	Multiplication	Expansion status	0
	Memory	Past activation status	Fixed value
APC	Antigen list	Strain IDs	Empty list

Table 3.3: Agent parameters and initial internal state

3.2.3.2 Desire-model specifications

Viral agents. These agents have a single objective: infecting a CD4 agent. This objective is permanent and is, therefore, also the initial objective. In the biological system, the objective is then to produce new virions, (implemented directly within CD4 agents).

CD4 agents. As long as a CD4 agent is neither activated nor infected, its single objective is to stay on "stand-by", ready to initiate immune responses. This is the initial objective and, since it does not correspond to any particular action, it can be limited to moving within the environment. The objective of an activated agent is to activate CD8 agents, while that of an

infected agent is to produce new viral agents. These two objectives can, of course, coexist. The objective of a CD4 agent which assumes the task of maintaining memory is similar to that of its initial state.

CD8 agents. As long as a CD8 agent remains non-activated, its single objective is similar to that of initial CD4 agents, and is implemented in the same way. The objective of a memory CD8 agent is similar. The objective of an activated CD8 agent is to multiply, and to eliminate target agents.

APC agents. An APC has two objectives: locating viral agents, and activating. The former is the initial state, and is similar to initial state of other immune agents. The latter can treated as is proposed for CD4 agents activating CD8.

3.2.3.3 Intention-model specifications

Viral agents, (Figure 3.2a, p.37). The raison d'être of a viral agent is species survival. This is reflected by the single objective: reproduce, i.e. infect. As infecting a CD4 agent is a permanent objective, every aspect of agent behaviour is targeted towards fulfilling this. Strategy is, therefore, simple, and is divided in three steps: the agent moves, inspects its new environment and, if possible, infects an immune agent. This is repeated until infection is successful or agent is destroyed.

CD4 agents, (Figure 3.2c). The initial objective being to explore the environment, behaviour is limited to agent movements. Once the agent is activated, each movement is followed by multiplication, (for a few iterations after activation), and information gathering on the environment, to determine whether there is a suitable target for activation. If such a target exists, activation follows. If infected, a new virion may also be produced. The behaviour of agent in "memory status" is again limited to agent movements.

CD8 agents, (Figure 3.2d). Initial and "memory" behaviour is similar to that of CD4 agents. Once the agent is activated, each movement is followed by information gathering on the environment, to determine whether there is a suitable target for elimination. If such a target

exists, it is eliminated.

APC agents, (Figure 3.2b). Behaviour of an APC agent has first divided in two steps: the agent moves, and then gathers information on its environment, in order to locate viral agents. Once it starts presenting details of such agents, it also looks for CD4 agents, in order to activate them.

3.2.4 Agent implementation

A summary is proposed in Figure 3.3, (p.38), of the four specific classes and single abstract class used for agent implementation. We introduced an early implementation of these classes in Perrin et al. (2006a;b). Here, we give a detailed and updated presentation of the attributes and methods. The former are used to implement an agent's internal state, while the latter are used for interactions, through several communication modes, (Section 3.1.2). In this case, commissive, expressive and declarative modes are not used. The other modes are required, and implemented.

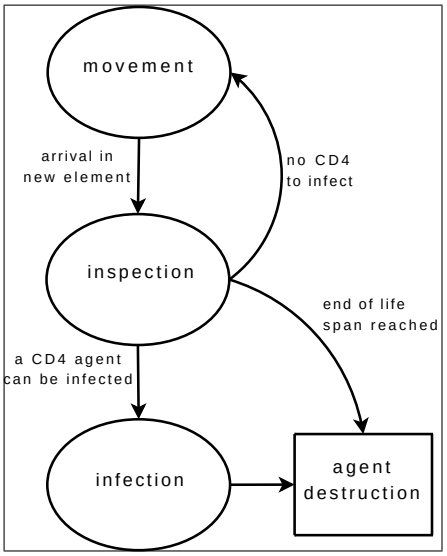
3.2.4.1 Base class

Base class attributes are related to the management of agent location and age. Direct implementation of the age would be ill-advised, as it would require updating age of all agents at each iteration, even when this information is not immediately required. This can be quite slow, especially as the number of agents increases.

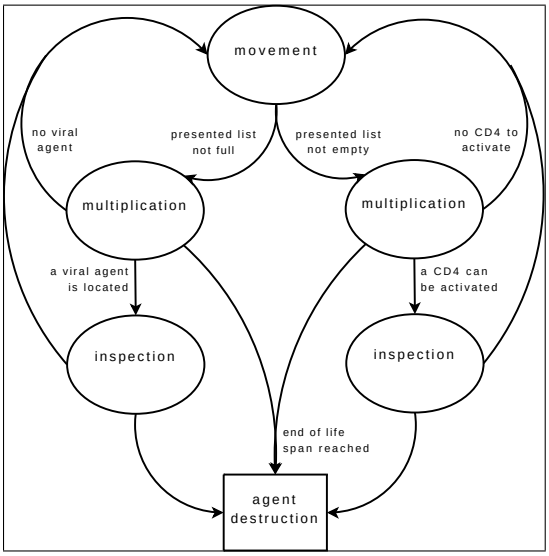
A more suitable alternative is to save the number of the iteration at which the agent was created. No repetitive update is required, and calculation of the difference between the current iteration number and the "birth date" of the agent provides its age when needed.

3.2.4.2 Viral agent class

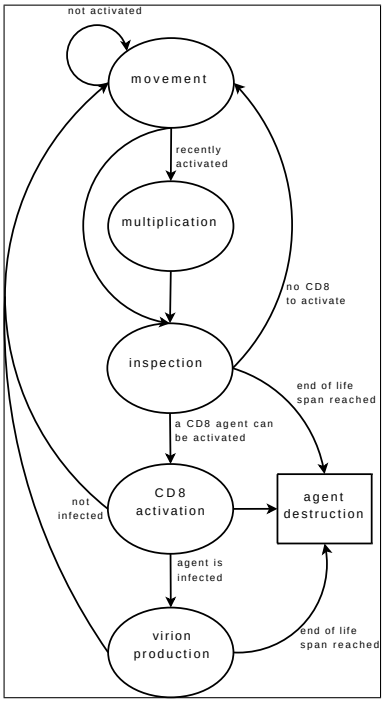
The viral agent class must manage viral strain information. Methods are, therefore, implemented to access and update this information, (i.e. representative and directive modes). Different viral strains mean different properties are needed for the associated viral agents:



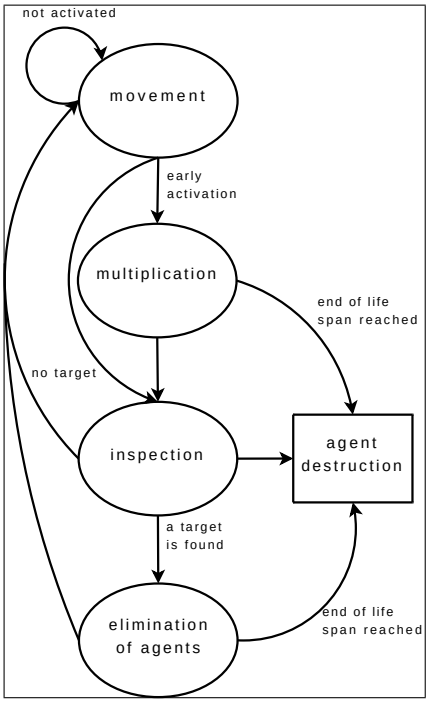
(a) Viral agent



(b) APC agent



(c) CD4 agent



(d) CD8 agent

Figure 3.2: Intention specifications: behaviour of the four types of agents

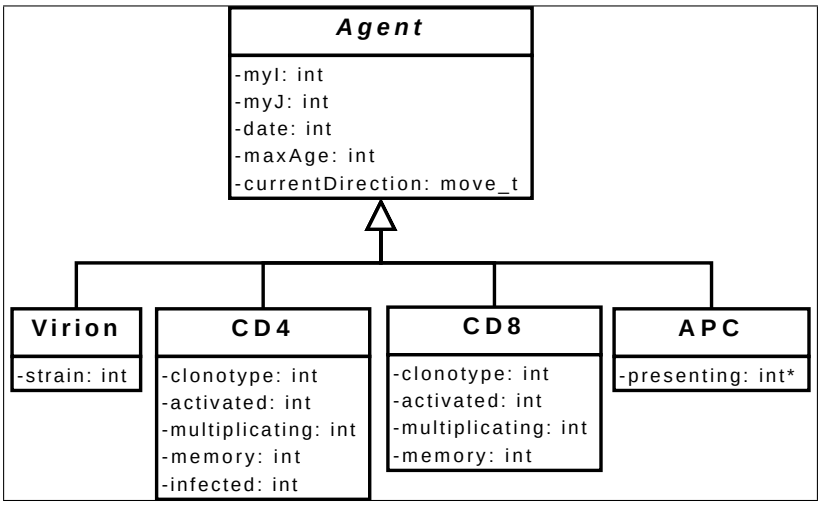


Figure 3.3: Agent implementation: class diagram

these are not recognised by same set of immune agents, may have distinct mutation rates, etc. Explicitly implementing all these properties within each agent would make them too "big", in terms of memory usage, (which is to be avoided for large simulations). The solution here is to use a single integer, coding the viral strain. This identification can then be used to access strain-specific properties, which are stored in a large array, (representing tens of thousands of potential strains).

An interesting property, here, is the identification of which immune clonotypes recognise each strain. As outlined, (Section 2.4.2), antigen recognition must be refined with respect to lock-and-key concepts. Proposed here is that two list of clonotypes should be available. The first list corresponds to clonotypes for which recognition is certain, (i.e. $p = 1$), and the second accounts for those for which recognition is not perfect, (i.e. $p < 1$). An important characteristic is that when an agent from the second list recognises the viral strain, the associated clonotype moves from the second list into the first. This is critical to the realism of the model, since it allows us to introduce some *adaptability*.

Infection of immune cells is implemented within CD4 agents. As this interaction involves only these two classes, this is biologically equivalent, but it is more efficient, in terms of computing performance.

3.2.4.3 CD4 and CD8 agent classes

Commonalities. Class attributes are related to management of the class-specific parameters that make up the internal state of these agents: agent clonotype, multiplication status, activation status, memory status, (and infection status for CD4 agents). Methods are provided, to update or access information, as for viral agents.

The clonotype is coded as an integer, randomly valued when the agent is created. Activation status is also coded as an integer. This value, set initially to zero, is used to store the identification integer representing the viral strain which led to agent activation. A positive value corresponds to an agent activated to respond to HIV strains, while a negative one is linked with another infection the immune system is currently responding to. This is crucial, as not all immune cells are available for a given response: some are already involved in other responses. It is also important, for CD4 agents, since infection requires that the cell is activated, but the response it is involved in need not be targeted to HIV.

Multiplication status is an integer, initially set to zero. It is then incremented at each step of the multiplication phase, until it reaches a limit and is set back to zero, signalling the end of this phase. Memory status is also coded as an integer and initially set to zero. When an agent starts assuming the role of a memory cell, it takes the value previously assigned for activation status, and that activation status is then set back to zero.

CD4-specific attribute values. Infection status is coded as an integer, initially set to zero, and storing HIV strain identification. Infection by multiple strains is possible in the real system. This is not, however, considered in the proposed model, as each set of strain properties may implicitly account for several strains in the real system, if these differ on properties which are not explicitly considered here. A possible extension of the model may be to consider multiple infections.

CD4-specific methods. Methods are implemented to account for modes related to infection of a CD4 agent. These correspond to assessment of viral presence in the neighbourhood, and transfer of viral content to the infected agent. Additional methods are also implemented for these agents, to deal with activation of a CD8 agent. The process is similar that of in-

fection of CD4 by a viral agent: presence of targets is assessed, and activation takes place if possible.

Similarly, other methods account for production of virions by infected CD4 agents. Here, the agent needs access both to the strain properties array and to its environment.

CD8-specific methods. Additional methods are also implemented for these agents, to account for elimination of infected agents. The process is similar that of previous interactions: presence of targets is assessed, and elimination takes place where necessary.

3.2.4.4 APC agent class

The APC agent class attributes are related to the management of the single parameter that makes up the internal state: the list of presented viral strains. Viral strains are coded using their identification integer. The list is initially empty, and new integers are added as strains are detected by the agent. Methods are provided to access and update this list.

Additional methods account for activation of CD4 agents. The process is similar that of previous interactions: presence of targets is assessed, and activation takes place where possible.

3.2.5 Multi-agent simulations of implemented model

3.2.5.1 Description of a typical iteration

Because of the nature of the system being studied, stochastic events are included, and the analysis of each configuration or set of parameters requires several simulation "runs". A key element for each of these is the update loop used to implement each iteration of the model. The time step used for these iterations in the implemented model is based on those mechanisms which have the fastest change, (e.g. production of a new virion by an infected CD4 cells, which takes on average just under a minute). The update loop is detailed below. Since all interactions take place within the physical space delimited by the matrix element, the *update sequence* of the matrix is not significant: it is not necessary to randomly select a different sequence at each time step. The focus here is, therefore, on updates at the agent

level.

Note: An interesting property of this update procedure is that not all agents are updated at each time step. This is to account for the fact, if all biological entities can move and interact at any given time, they do not do so on a continuous basis, nor on a regular rhythm shared by all. This pattern is implemented through uniform random selection, within a matrix element, of agents that are updated at the current time step. Due to the finite size of the matrix element, an agent will not spend an infinite time without being updated. Moreover, since the first action when selecting an agent is to check its age, it also guarantees that no agent will have an effect outside of its life span, even if not updated for some time. Finally, while each agent type is responsible for a given type of interaction, all are also passively involved in other interactions. For instance, CD4 agents coordinate CD8 activation explicitly, but are also implicitly updated when they are involved in their own APC-controlled activation.
This random selection of agents involved is a significant advantage, as it is both biologically realistic and computationally efficient, (due to reduction of operations required for each update iteration). An important consequence is that it allows larger agent populations to be modelled, (a key objective).

Update of viral agents:

1. *Movement*. An agent is selected within the current matrix element, and moves to a new matrix element.

2. *Query*. The agent gathers needed information on this new environment, i.e. number of CD4 agents.

3. *Infection*. Providing there are such agents, one is randomly selected. (This accounts for the fact that not all agents within that small space are considered to be in physical contact with this viral agent). If this selected CD4 agent is a suitable target, (i.e. activated and not previously infected), an infection process starts: value of attribute "infected" of this agent is set to value of attribute "strain" of viral agent, (or to a

related value if successful mutation occurs[4]), and the viral agent is then destroyed. If no suitable target was accessible, nothing else happens.

Update of CD4 agents:

1. *Memory*. When an active CD4 agent reaches the end of its life span, it may become a memory cell, which is implemented as follows: value of "memory" is then taken from "activated" attribute, which is set back to zero.

2. *Movement*. The agent moves to a new matrix element. If it is not activated, nothing else happens.

3. *Query*, (presence of CD8 agents if activated, presence of "reactivation agents" if non-active memory cell).

4. *Reactivation*. If the CD4 agent is a non-active memory cell, it can be reactivated if it locates infected agents or agents involved in immune responses, and recognises viral strains associated to these. If the CD4 agent is not reactivated, the update stops here.

5. *Expansion*. If the CD4 agent is currently multiplying, a clone agent is produced, (the only distinct parameter corresponding to age of the new agent). If the CD4 agent is not infected, the update stops here.

6. *CD8 activation*. Providing there are CD8 agents, one is randomly selected and, if it is not already activated, clonotypes are compared. If these are compatible, activation occurs: value of attribute "activated" of CD4 is copied into the respective attribute of selected CD8 agent.

7. *Infection*. A new virion may be created, based on properties stored in the strain array, and accessed using value stored in "infected" attribute.

Update of CD8 agents:

1. *Memory*, (as for CD4 agents).

[4]If mutation is unsuccessful, infection fails, but the viral agent is still eliminated

2. *Movement*. The agent moves to a new matrix element. If it is not activated, nothing else happens.

3. *Query*, (presence of targets if activated, i.e. infected CD4 agents and presenting APC agents, presence of "reactivation agents" if non-active memory cell). Possible *reactivation*, (as for CD4 memory agents).

4. *Reactivation*, (as for CD4 agents).

5. *Expansion*, (as for CD4 agents).

6. *Elimination*. If targets are located, they are randomly selected and eliminated.

Update of APC agents:

1. *Movement*. The agent moves to a new matrix element. If it is not activated, nothing else happens.

2. *Query*, (presence of viral agents, and presence of CD4 agents if already presenting antigens).

3. *Antigen acquisition*. If viral targets are present, one is selected, its "strain" attribute added to the list, and the agent destroyed.

4. *CD4 activation*. If the list is not empty and if there are potential CD4 targets for activation, one is selected and activated, through a copy of a "presenting" integer from the list into "activated" attribute of target.

3.2.5.2 Random number generation

Many aspects of a real-life system involve stochastic events, and, consequently, most methods and functions in our model have to include random number generation. Details on stochastic aspects of the immune system are well-reported, (see e.g. Germain (2001)). Examples include the process by which new lymphocytes are created: a lymphocyte can only recognize a specific set of antigens so that, to protect itself against any attack, the body

has to generate thousands of "variations" between lymphocytes. This is implemented using random numbers. Further, one of the most distinctive features of the virions is their high mutation rate, which occurs randomly. Finally, there is no sensible way to deal with mobility unless we include stochasticity.

A full-scale model will involve millions of agents in very long simulations, with parallel aspects involved. A reliable and efficient random number generator which can deal with all these stochastic elements and features is essential. Many generators are, of course, available, and good ones can also be designed explicitly (see e.g. Press et al. (2002)). A top-quality parallel generator is needed, and our model can interface both with the Scalable Parallel Random Number Generators library, (SPRNG), (Srinivasan et al., 2003) and with the "Mersenne twister" generator (Matsumoto and Nishimura, 1998). These libraries incorporates recent, state-of-the-art developments in the mathematics and computer science of parallel pseudorandom number generation. They both have an existing, active user base, ensuring high standards. In particular, they allow the streams to be also absolutely reproduced, for computational verification, independent of the number of processors used in the computation. High confidence in the statistical results, at a very low computing cost, is a feature of this library usage.

3.3 Chapter summary

In this Chapter, detailed implementation of agent structure is presented. Four types of agents are used, in order to fully describe cell-mediated response and HIV mechanisms, while reducing agent diversity to a level which permits simulation of large populations. The proposed implementation, therefore, provides a basis from which to address the limitations outlined in Chapter 2.

Implementation of agent classes, as well as typical iteration of simulation, are detailed. This outlines several features that were also presented as essential in that Chapter:

- Sensible granularity is obtained by using a time-step of fifty seconds, which ensure

that no significant interaction is unaccounted for.

- Structure of CD4 and CD8 agents allows direct cell-level control of immune memory. This is expected to be more realistic than a centralised implementation.

- Antigen recognition is refined to include adaptability: recognition is not binary, and this essential feature is also expected to enhance model realism.

The following Chapter focuses on implementation of the lymph network and details how this can address limitations outlined for previous models.

Chapter 4

Building the lymph network

4.1 Structure of the lymph network

Objectives detailed in Chapter 2 include explicit implementation of lymph nodes and the lymph network. In the next two Sections additional details on these structures and a further demonstration of their essential role in achieving model realism, are provided.

4.1.1 Lymph nodes

Lymph nodes are key components of the lymphatic system, and are found throughout the body. An important function of a lymph node is to act as a "filter" for foreign particles such as viruses, by collecting antigens. Even more crucial, however, is that this "filtering" is used to enhance immune response. A node's structure is designed for this function, and is characterised by high content of immune cells, which use these lymph nodes as meeting points for large-scale responses that could not be mounted in a less favourable environments.

The normal size of lymph nodes ranges from a few millimeters to around 1-2 centimeters, (see e.g. Tiguert et al. (1999)), but, as lymphocytes multiply during activation of the immune response, nodes can become significantly enlarged during such periods.

The inner structures of a node, illustrated in Figure 4.1 (reproduced from Gray (1918)), are associated with a predominant cell in each case: *outer cortex* is rich in B lymphocytes,

while *deep cortex* and *medullary cords* predominantly host T lymphocytes, and *medullary sinuses* can contain some immobile macrophages.

A node is connected with the "outside" through blood and lymph circulations.

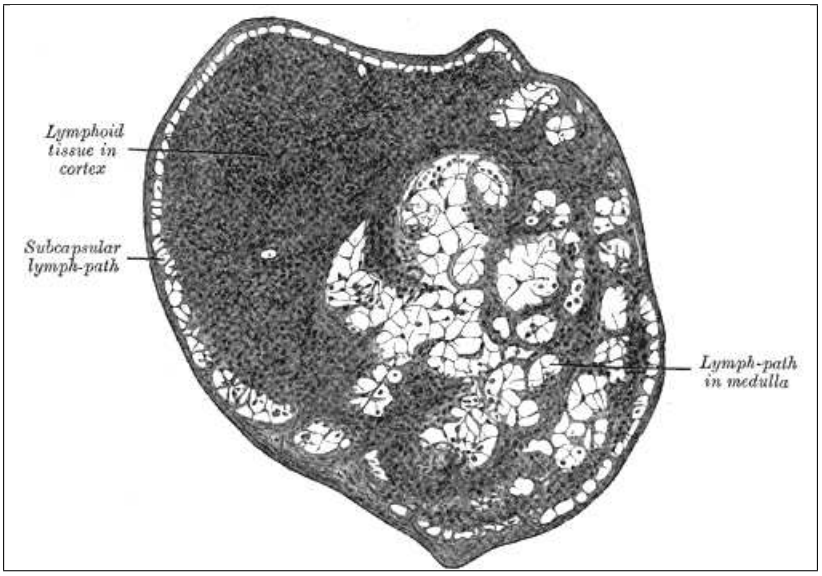


Figure 4.1: Section of a lymph node, (reproduced from public-domain book (Gray, 1918))

4.1.2 Distribution and circulation

As key defense units, lymph nodes are distributed throughout the body. Humans have around 500 lymph nodes, and lymphocytes constantly circulate through these lymph nodes. To guarantee this cell circulation, connectivity is an essential property of the lymph network: a cell newly produced in the thymus must be able to reach any lymph, and efficient immune response implies interactions between nodes, (by means of cell exchanges).

The lymph network, however, is *not* equivalent to a *complete graph*: between any given pair of nodes, there is a path, but not necessarily a direct connection. In contrast, the lymph network is organised as a set of chains. These "clusters" of nodes can be found in the neck, chest, abdomen, underarm, etc. For illustrative purposes, lymphatic chains of head and neck, and of stomach, are shown in Figures 4.2 and 4.3, respectively, (reproduced from a public-domain edition of *Gray's Anatomy*, (Gray, 1918)).

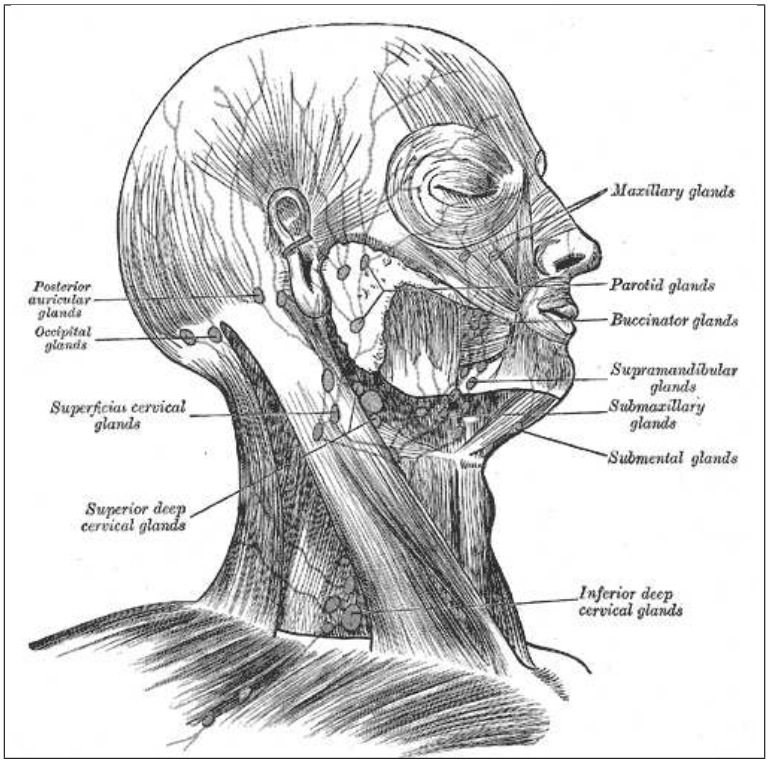


Figure 4.2: Lymph nodes of head and neck, (reproduced from Gray (1918))

The circulation of immune cells between nodes is not trivial. Cell migration is non-random (Witherden et al., 1990), and is, in particular, type-dependent: CD4/CD8 ratio is always higher in the recirculated compared to overall population, (and often more than twice as important). Interestingly, however, transit kinetics are equivalent for both subsets of T cells. Naive and memory cells also have different recirculation pathways (Mackay et al., 1990).

Circulation is also tissue-specific, in the sense that T lymphocytes preferentially recirculate back to the tissues they came from. This is controlled at the molecular-level through specific chemokines (Kunkel and Butcher, 2002).

As might be expected, circulation patterns are affected by immune response. In particular, lymphocyte input and output is significantly increased in nodes where immune activation is taking place (Cahill et al., 1976).

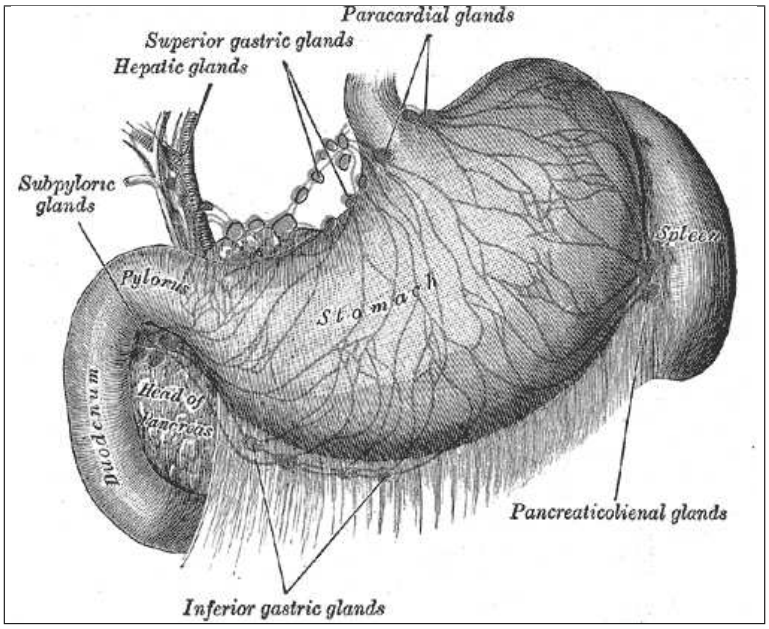


Figure 4.3: Lymph nodes around stomach, (reproduced from Gray (1918))

4.2 Importance of lymph nodes and lymph network in the context of HIV infection

Lymph nodes are essential units, since they host the most crucial part of immune response to any infection. An immediate consequence of this is that any immune model, not explicitly dealing with these nodes, is neglecting most of the immune response it is trying to account for.

This remains true, of course, in the context of HIV infection, and in many respects, the lymph system is even more crucial in this particular context. In recalling that HIV replication cycle is based on infection of CD4 cells, lymph nodes are precisely where most of these immune cells migrate to. Recirculation of course implies that some also are found in blood vessels, for instance, but at any given time the majority of CD4 is located still within lymph nodes. A model without explicit lymph node implementation fails, realistically, to account both for the immune response and for the infection this response is targeting.

Moreover, dynamics of cell mobility are known to be altered by HIV infection (Douek et al.,

2003). This is a direct consequence of generalised immune activation, which is known to affect cell recirculation. In particular, an inversion of the CD4/CD8 ratio is observed in peripheral blood, due to diverging population dynamics for these cell types: $CD8^+$ cells undergoing rapid expansion through immune activation are not subject to viral infection. This has led to reports that decline in peripheral blood $CD4^+$ cells overestimates actual cell loss in lymphoid tissue (Rosok et al., 1996). However, more recent evidence suggests that, on the contrary, total-body depletion is underestimated by peripheral blood cell counts (Douek et al., 2003). Whether these counts under- or overestimate actual effects is (almost) irrelevant. More crucial is the fact demonstrated that these counts are not an accurate measure of fitness of the immune system, and that an explicit model is required.

HIV infection also affects node structure, (through chronic activation and cell depletion, see e.g. Biberfeld et al. (1985)), and cell dynamics, (through altered balance between naive and memory cell (Roederer et al., 1995), and increased pressure on thymic output Douek et al. (1998)). An explicit implementation of the lymph nodes can permit inclusion of this structural distortion.

4.3 Node structure and implementation

The proposed model provides a solution to explicit implementation by using lymph node as the key abstraction element for the "world" in which agents evolve. Details are provided in this Section.

4.3.1 Matrix-based representation and associated neighbourhood

In order to efficiently account for cell interactions and corresponding physical contact, accurate localisation of agents within the lymph node is required, implying division into smaller subunits. A standard approach to spatial representation requirements is to use a 2D or 3D matrix, and to consider either a Von Neumann neighbourhood, or Moore neighbourhood, (Figure 4.4).

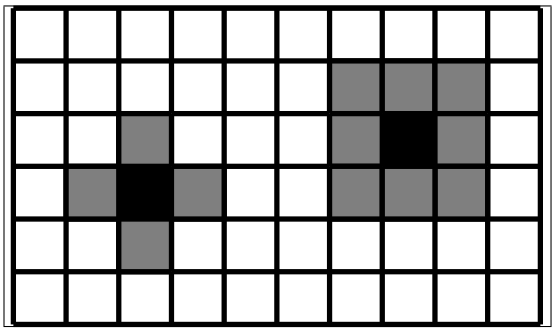


Figure 4.4: Von Neumann (left) and Moore (right) neighbourhoods

Division into subunits is not trivial however, and the standard approach does not appear particularly suited for multi-agent systems involving several types of agent. In our new model, each matrix element has to be able to contain at least one agent of each type. If it can contain no more than one, limitations related to size of the matrix and of the modelled entities are an issue. In the former case, modelling a million viral agents requires at least a 1000x1000 matrix; in the latter, not all biological entities involved have a similar size. It is, therefore, unrealistic to assume that a matrix element should contain the same number of viral agents, (small entities), and APC agents, (significantly larger cells).

More usefully, matrix elements can be considered to be "physical", containing several agents of each type, with a limit based on size of modelled entity. This implies that all cell interactions will take place within this physical neighbourhood, and considering surrounding matrix elements is not necessary. In our model, lymph nodes are, therefore, implemented as 2D matrices where each element represents a physical, 3D, neighbourhood.

4.3.2 Memory allocation and agent localisation

A naive implementation of agents within the matrix representation of a lymph node would allow for arrays of agents within each element. Each array would store agents of a given type and localisation of agents would, therefore, be trivial, with agent movement between elements involving simple deletion from one array and addition to another.

Memory allocation for agents also offers an intuitive solution: memory could be dynamically allocated whenever an agent is created, and freed when one is destroyed.

Both these solutions would be suitable for a small-scale system, but would not be efficient for a large model such as the one studied here. In the first place, memory allocation is one of the slowest operations on a computer, and would be intensively solicited here. At each iteration, a large number of agents are created or destroyed, (Table 3.2, p.32). Successions of dynamic allocations and deallocations would be used and would, therefore, significantly hinder efficient computation. Moreover, as immune response is initiated, agent count will sharply increase, and may reach values close to theoretical limits of the model. The main advantage of dynamic allocation, (i.e. using only as much memory as is needed at any time point), would not apply in such a simulation.

Consequently, our implementation is based on static memory allocation. Large arrays of agents, (one for each agent type), are allocated when simulation starts, and store the maximum number of agents which can be present in the whole matrix at the same time. Each matrix element then only stores integers, used as offsets, to locate agents currently held in these arrays.

4.4 Cell mobility and validation of the lymph node model

4.4.1 The importance of cell mobility

Cell mobility is often neglected in existing models, in part, due to lack of explicit implementation of lymph nodes. Nevertheless, consideration of mobility is essential, especially for a bottom-up approach, as accurate accounting for low-level mechanisms is the aim. The main focus is on cell-level interactions, with results based on cell-dependent information. For instance, (see Chapter 2), affinity between immune clonotype and viral epitope determines efficiency of immune response, and physicochemical binding between viral glycoprotein gp120 and CD4 receptors of lymphocyte is a first step in cell infection.

Some models try to account for physical contact in these interactions by probability state-

ments on proximity of cells, but there is no biological basis for such a function to be used, and it seems unlikely that a single constant probability would be valid throughout the body. The agent-based paradigm offers the opportunity to fully account for cell mobility, both within and between lymph nodes.

4.4.2 Intra-node mobility

Implementation

Intra-node mobility refers to agent movements within each matrix. Overall movement is based on flow, created by the fact that agents enter the node on one side, and exit on the other. The required distribution of agents and time spent in each node is stochastically governed.

Using the implementation and memory allocation strategy detailed in Section 4.3, updates of agent localisation are easily managed: (i) creation of an agent requires initialising its internal state and saving the offset, to access this state, in the matrix element; (ii) deletion of an agent involves deletion of this offset; a new agent can subsequently be created in this space; (iii) movement from a one matrix element to the next requires transferring the offset value to the destination element.

Validation

Intra-node mobility is validated using a "covering test". Generic agents are created, (i.e. biological interactions are not taken into account here), and are randomly scattered over a matrix. These movements are then monitored over a number of iterations, to assess whether all matrix elements are indeed accessible. Tests were performed with a single agent in a 50x50 matrix, and with ten agents in a 200x200 matrix, and results are shown in Table 4.1. Even with limited iterations (50,000, as opposed to several millions for a full-scale simulation) and few agents, covering is very satisfactory and supports adequate modelling of intra-node mobility. Interestingly, this model feature is implemented at a very low comput-

ing cost, since agent movement only involves alteration of two integer values. It does not, therefore, limit other modelling objectives such as efforts towards simulation of large agent populations.

Matrix size	Visited elements	Maximum	Minimum	Average	Std. deviation
50x50	2500 (100%)	103	3	39.94	15.45
200x200	39795 (99.49%)	108	0	24.98	15.14

Table 4.1: Covering tests on intra-node mobility

4.4.3 Validation of cell-level mechanisms in the lymph node model

The lymph node structure is fully implemented, and populated with agents which incorporate mobility features. The immediate objective is to validate this structure, before subsequent validation of the lymph network model, which uses this node model as key abstraction unit, (Figure 2.4, p.23).

The first stage is to guarantee a high temporal granularity, which is essential to model realism (Section 2.4.2). A time step of just under a minute has, therefore, been chosen for this lymph node model, which addresses the limitations of the long time-step models and allows accurate simulation of cell-level mechanisms. Tests to validate these basic interactions are presented here.

The difficulty is that there is no simple criterion, such as a convergence rule, since the system is continuously evolving. The only solution here is to run a significant number of simulations, acquire sufficient clinical data (e.g. Buseyne and Riviere (2001); Murali-Krishna et al. (1999); Oxenius et al. (2001)), and isolate similar patterns in both sets: this motivates the following tests.

A first consideration is to validate the activation of the effector cells of the cell-mediated response, i.e. CD8 lymphocytes. In the biological system, we highlighted three steps:

1. CD8 cells are activated by CD4 cells to target a specific antigen;

2. newly-activated CD8 cells multiply themselves and “attack” suitable targets;

3. once the response is over, a portion of the created cells become memory cells, while the others are destroyed.

To assess how the lymph node model deals with the biological system requirements, a particular configuration is simulated with no viral agents, but a set of artificially activated CD4 cells. These agents activate their CD8 counterparts, which rapidly multiply themselves. Once the response is seen as over, (no further targets), most excess CD8 cells are destroyed while a few are kept as *memory* cells. Observed patterns are satisfactory, (see Figure 4.5).
Next, consideration is given to the three steps leading to activation of CD8 cells in the cell-mediated response:

1. APCs detect foreign entities and start presenting specific antigens on their surface;
2. given CD4 lymphocytes recognize these antigens and activate themselves;
3. these cells activate CD8 lymphocytes, which then behave as previously described.

To assess the lymph node model in this case, "passive" viral agents are simulated: these move within the node and can be recognized as foreign entities, but do not show any HIV-specific behaviour. This generates results, which are focused on the response itself. In Figure 4.6 (p.57), the chain of activation is reproduced. The delay between each activation step is due to the mobility condition: cells need physical contact to interact.
The last validation step for local interactions is the HIV-specific behaviour that viral agents have to reproduce. HIV virions use activated CD4 lymphocytes as hosts. Once a cell is infected, it starts producing new virions, while its life expectancy is greatly reduced, even without the consideration that it may be destroyed by activated CD8 lymphocytes.
This specific behaviour was highlighted in a famous experiment conducted by a research group led by Montagnier, (Barre-Sinoussi et al., 1983). It was thought at the time that the first human retrovirus discovered, HTLV-1 (Human T cell leukemia virus), was responsible for AIDS. HTLV-1 is known to make T cells capable of indefinite growth and division. A set of CD4 T lymphocytes was introduced to biosamples from a patient suffering from early

symptoms of AIDS. Some retrotranscriptase activity was detected, but soon disappeared. A new set of lymphocytes was introduced into the culture: activity was again observed initially, then disappeared. The virus responsible for AIDS cannot, therefore, be HTLV-1, as the former "kills" CD4 cells.

The lymph node model must reproduce this expansion strategy and its success has been assessed through a simulation involving only virions and CD4 agents, and featuring a massive input of "fresh" CD4 agents at some stage. As can be seen, (Figure 4.7), for the new set of lymphocytes introduced after 750 iterations, a decline in CD4 is again observed and the test suggests that the model satisfactorily reproduces *in vitro* behaviour. Visualisation[1] of expansion strategy is shown in Figure 4.8, (p.58).

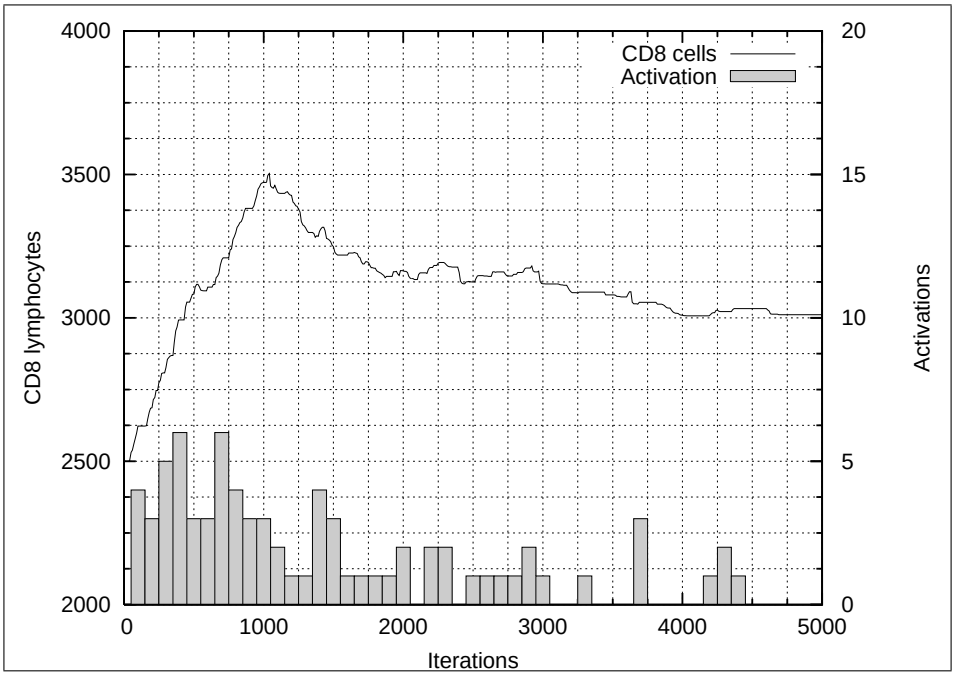


Figure 4.5: Activation and multiplication of CD8 lymphocytes

[1] Visualisation was only available for early single-node tests, (see Section 8.2 for details).

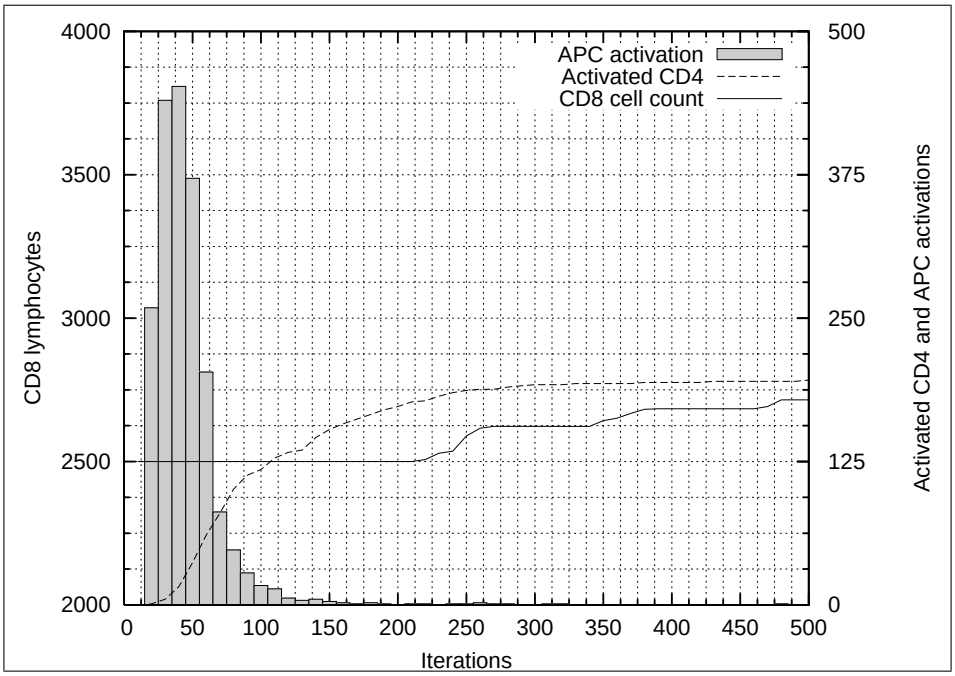


Figure 4.6: Chain reaction leading to the cell-mediated response

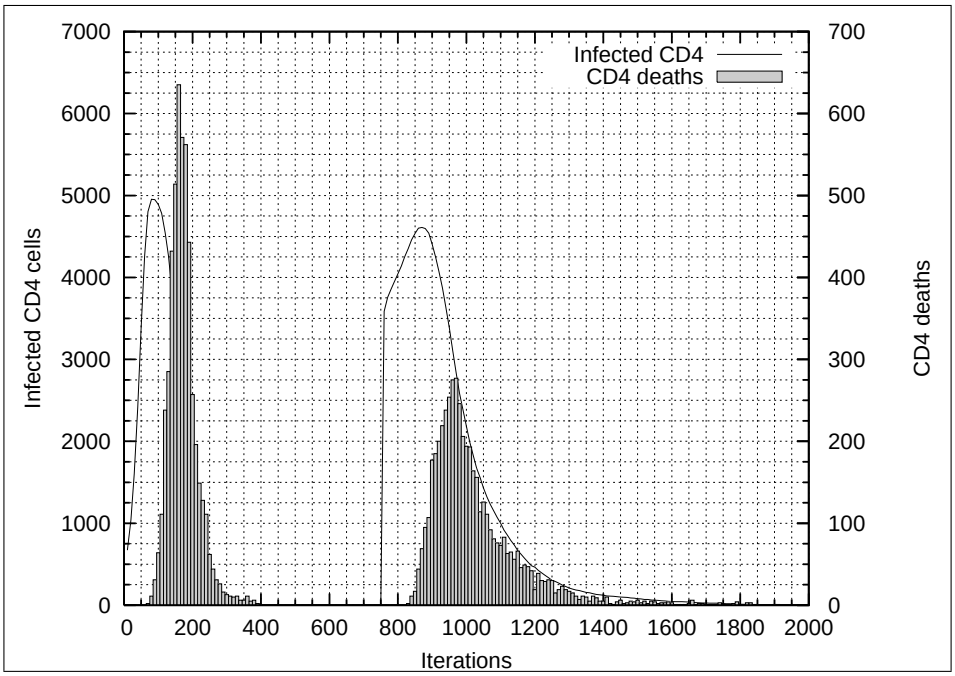


Figure 4.7: Viral agents using Th cells as hosts, leading to cell depletion

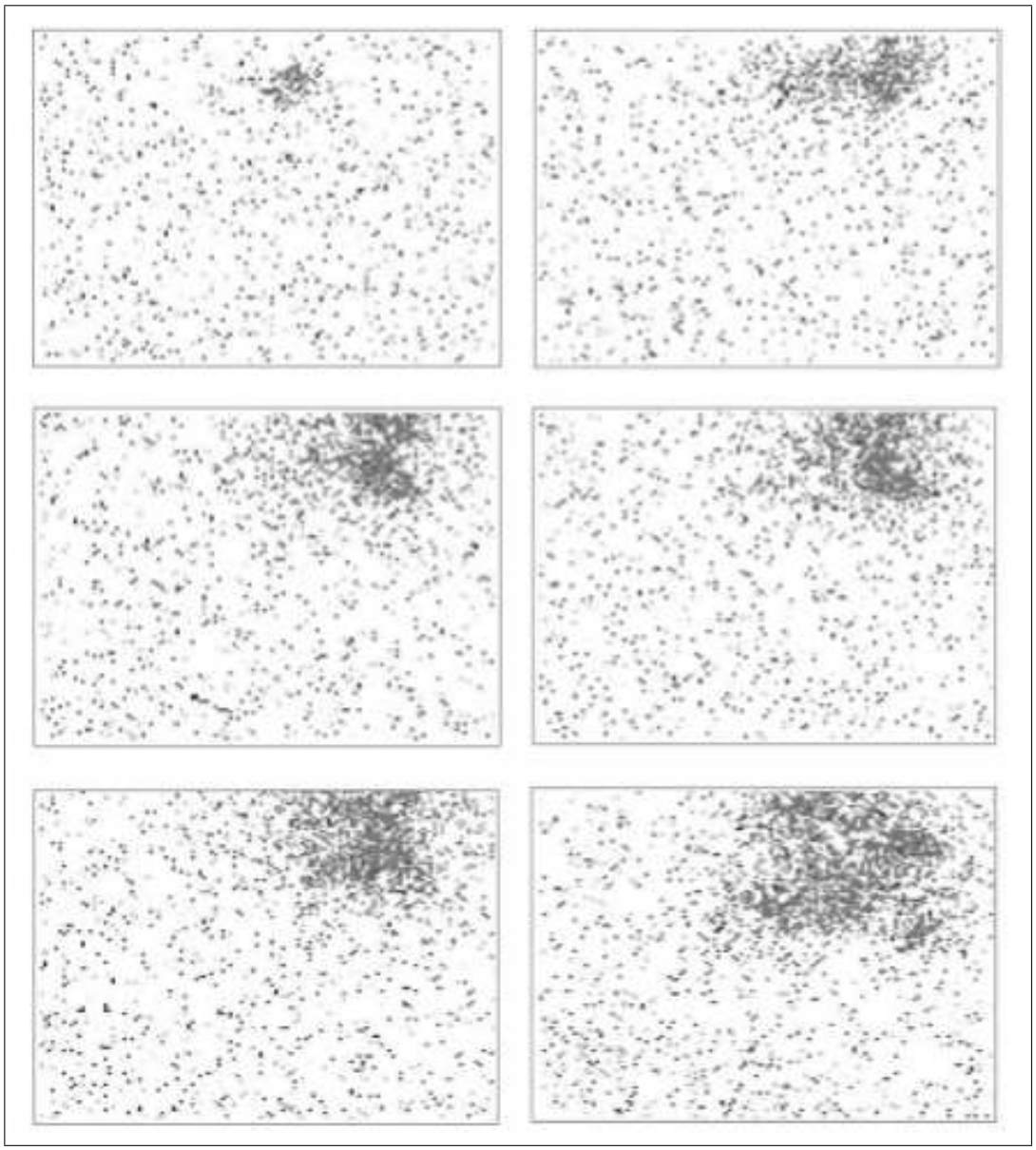


Figure 4.8: Graphical representation. Infected CD4 agents, (black dots), are producing new viral agents, (dark grey dots).

4.4.4 Inter-node mobility

Concept

Cell movements also involve *inter-node* mobility. This refers to agents leaving a lymph node entirely and entering another one. As far as node structure itself is concerned, this requires addition of entry and exit points to the matrix, for agent arrivals and departures. When an agent reaches the exit at the far corner of the matrix, it is added to a transfer list, and then removed from the node. At the end of the iteration, lists are shared and agents transferred to their new locations.

Structure

It is important to account for lymph network structure: this, (as mentioned earlier), is similar to a directed connected graph, but is not complete. It implies that the final destination of an agent leaving a node can theoretically be any node, but its immediate destination is limited to a small subset of nodes.

Final destination

Selection of final destination is random, but not based on a uniform distribution. Lymphocyte recirculation features preferential recirculation back to tissues of immune cell origin. To account for this, each lymphocyte agent leaving a node has a 0.5 probability to be assigned this node as its final destination, ensuring circulation through the whole lymph chain corresponding to this node. To respect the higher lymph node to overall population CD4/CD8 ratio in departing agents, a function is also added to guarantee that not all CD8 agents evolving in the neighbourhood of the exit point are selected for departure.

Immediate destination

Selection of immediate destination is based on the lymph network structure modelled. The number of lymph nodes modelled during a given simulation may vary, for instance de-

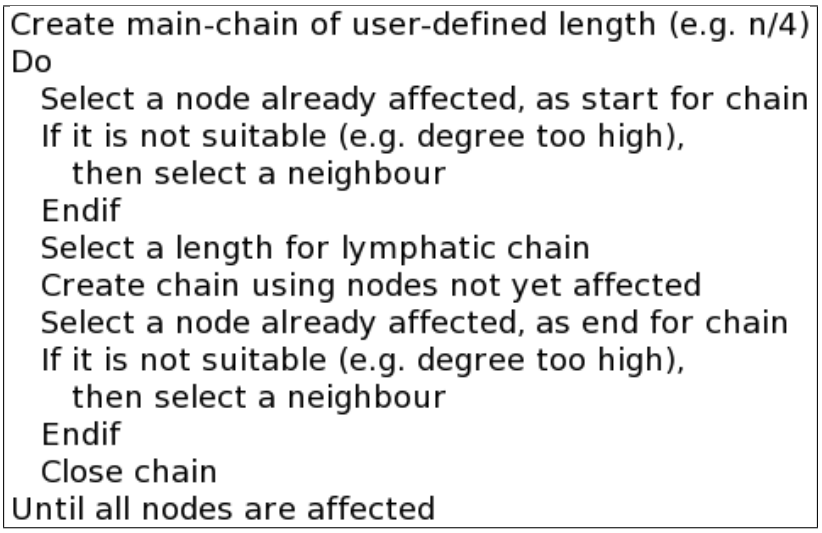


Figure 4.9: Principal algorithm for lymph network generation

pending on available computing resources. What is needed here, therefore, is an automated technique to generate a network of nodes reproducing the lymphatic chain structures for any number of nodes. A generation method has been developed, implemented and tested, and the principal algorithm is shown in Figure 4.9. It is based on creation of several lymphatic chains, which are linked together by a "main-chain" to ensure connectivity of resulting structure. An example of a generated network is shown in Figure 4.10.

Inter-node mobility, of course, implies simulations with a significant number of nodes, and this implies increased computing load. A parallel implementation, (detailed in the next Chapter), is employed to address this final requirement. The results of inter-node mobility, in conjunction with parallelisation features, are detailed in Chapter 6.

4.5 Chapter summary

In this Chapter, it was demonstrated that an explicit implementation of lymph nodes and associated network is both necessary and achievable. It is necessary, due to the organisation of the immune system, where lymph nodes are key defense units which host most of the immune response to HIV. Furthermore, HIV uses immune cells as hosts to replicate and

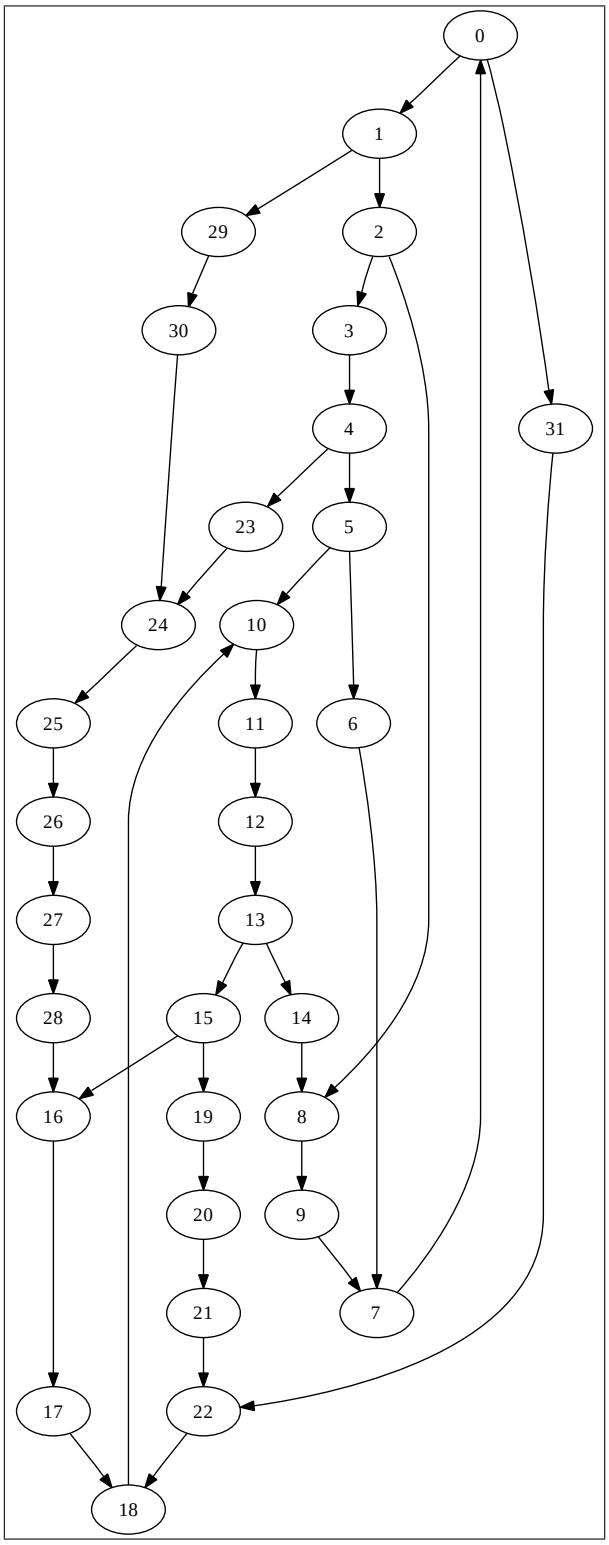


Figure 4.10: Example of generated lymph network structure with 32 nodes

proliferate, and it is in lymph nodes that the highest concentration of these is found.

The generic nature of the agent-based paradigm allows agents to evolve in any user-defined environment, and an implementation is proposed here, which explicitly implements lymph nodes and lymph network structure, and accounts for cell mobility, both within lymph nodes and between them. As such, it addresses an essential feature, (Section 2.4.2).

Intra-node mobility was obtained at very low computing cost, for which the implementation does not limit modelling of large agent populations, while tests on different matrix sizes have shown that behaviour is satisfactory. Inter-node mobility was implemented by adding entry and exit points to matrices, and probability functions to determine the final destination of circulating agents. The immediate step on the path to that lymph node is obtained through introduction of an algorithm generating lymph network structure.

Four of the six limitations identified, (Section 2.4.2), have been addressed satisfactorily. An efficient parallel implementation of the lymph network is also required to fully account for inter-node mobility and large-scale, biologically-meaningful simulations. Details on this are given in the next Chapter.

Chapter 5

Parallel Implementation

5.1 Rationale for a parallel implementation

5.1.1 Achieving realism: one is not enough

Since most of the immune response to HIV infection takes place within lymph nodes, these key organs were chosen as the basis for the agent-based model.

Mechanisms for ensuring physical contact with respect to immune interactions generated a discussion on mobility, and the chosen implementation was detailed in the previous Chapter. Cell mobility within each node can be examined with a single-node simulation, as movements are local. Mobility between nodes, however, implies that nodes must be connected via a *network*.

Another motivation of this work, identified in Chapter 1, is to look into localised effects such as early infection in the gastro-intestinal tract. This again requires the implementation of an additional layer, or network, that changes local node properties, (Chapter 6). In order to create these subnetworks, i.e. areas with distinct behavioural patterns, there is, thus, a need for a lymph network large enough to accommodate both "normal" and "localised" nodes.

The approximately five hundred nodes mimicking those of the human body can not be handled on a single-processor computer, because of limitations in both available memory, (it is

not possible to allocate a large number of nodes), and computing power, (it would take too long to simulate all nodes). The only solution is to consider a parallel implementation of the lymph network.

5.1.2 Parallel nature of the problem

When considering the parallel nature of a problem, classification is built on three categories, (see e.g. Gropp et al. (1999b) for details):

- *Embarrassingly parallel problems*, which can be broken down into subparts, each completely independent of the others. As a consequence, no communication is required, except to split up the problem during initialisation and to combine the final results at the end. In such cases, *linear speedup* can be expected from a parallel implementation. A well-known example is Monte-Carlo simulations.

- *Regular and synchronous problems*, in which the same instruction set, (regular algorithm), is applied to all data with synchronous, or loosely synchronous, communications, with each processor finishing its task at the same time. These usually require local, (neighbour to neighbour), and collective communication, the latter being used to combine final results. As long as the ratio of computation to communication is large, parallel implementations for these problems provide almost linear speedup for local communication, (and is slightly less efficient for non-local communication). Examples include Fast Fourier transforms (synchronous), matrix vector products and sorting (loosely synchronous).

- *Irregular and/or asynchronous problems*, characterised by irregular algorithms which cannot be implemented efficiently except with message passing and high communication overhead. In such cases, communication is usually asynchronous and requires careful coding and load balancing. Dynamic repartitioning of data between processors is often required. Speedup is, therefore, difficult to predict, but should not be expected to be linear, or even close to. Any moving boundary simulation typically

falls into this category.

The model proposed here uses the lymph network as the "world" in which agents evolve. As there is no direct interaction between cells located in separate lymph nodes, (Chapter 4), cell-level interactions require physical proximity. The consequence is that, *apart from cell transfer*, each node is independent of the others. A node influences its neighbours solely through cells exiting their current location and reaching the next node.

Local interactions have been implemented by a regular algorithm, (detailed in Chapters 3 and 4). Since node size is constant, there is little variation in the agent counts of the various nodes, and we can expect local iterations to finish around the same time. Our problem is thus regular and loosely synchronous, and the model is suited for a parallel implementation.

5.2 Challenges and implementation choices

5.2.1 Parallelisation: *Divide et impera*

The main principle of program parallelisation can be found in the famous Latin proverb *Divide et impera*, "Divide and conquer". Identification of possible "divisions" within the program is essential. In some programs, a set of instructions is repeated several times, with each iteration independent of the others. Monte-Carlo simulations are a typical example here. In such cases, several iterations can run at the same time, using a *time parallelisation*. This technique can not be applied to the lymph network model however: each iteration i uses the final state of the agents after iteration $i-1$, so that two iterations can not run at the same time. Yet, as highlighted in the previous section, the problem does have a parallel nature, in the sense that each node is largely independent of the others.

We can, therefore, use a *spatial parallelisation*, based on a reciprocity between the lymph nodes and the computer nodes of a cluster. Each lymph node of the model is assigned to a computer node of the parallel architecture, and communication network is designed to mimic cell mobility along the lymph network.

5.2.2 Expected difficulties

Spatial parallelisation has previously been investigated, e.g. in the context of Monte-Carlo simulations, for HIV infection (Hecquet et al., 2007), with the main disadvantage in that case being the communication overload. The parallelisation strategy, detailed above, guarantees that communication is kept to a minimum. Indeed communication between computer nodes is only used to represent actual cell exchanges; we do not add any communication overhead due to the parallelisation itself, apart form initial problem splitting and final results gathering.

This is an important point: as communication is often seen as the bottle-neck of any parallel implementation. This is usually a consequence of the hardware architecture, where physical data transfer is significantly slower than computing operations. As shown in Equation 5.1, data transfer time depends on data size, of course, but also includes a fixed *network latency*[1].

$$TransferTime = Latency + DataSize/Bandwidth \tag{5.1}$$

Over the last decade, efforts on parallel hardware architectures have led to a reduction of the network latency, which is now generally less than 100 μs (Mamidala et al., 2006), and sometimes advertised to be as low as 1.5 μs (Pathscale^{TM}, 2005). This needs to be put in context with current computing speeds, where recent configurations reach hundreds of teraFLOPS[2], i.e. in that context, a millisecond represents a million basic operations. Clearly, therefore, no unnecessary communication should take place if avoidable. The choice outlined, (Chapter 4), is to use a single list for agent transfer, (rather than as many lists as agent types).

For biological models, a balance must be found between minimal communication and need for realistic exchange of information, and this concern motivates much of the remainder of this chapter. Firstly, in what follows, we introduce the communications protocol used for

[1] Network latency corresponds to the time delay between initiation of the communication and actual start of data transfer.

[2] FLOPS: FLoating Operations Per Second.

implementation.

5.2.3 Implementing: MPI

In brief, MPI[3] is a communications protocol used for parallel implementation of programs. MPI provides support for point-to-point and collective communications, enquiry routines to query the execution environment, as well as constants and data-types (Gropp et al., 1999b;a).

MPI is a low-level library, which provides an interface to C, C++ and Fortran 90. MPI is, therefore, a language-independent protocol. Portability was a priority during its development, and MPI is platform-independent. This is decisive in the context of the study presented here, as the model is executed on various parallel platforms.

This, along with high performance and scalability, made MPI the *de facto* standard and most current distributed-memory computers offer MPI implementations. MPI is taught and used widely, which, together with the availability of open-source implementations and the large body of programs that require MPI, (including both Research models and commercial products), guarantees long-term legacy of MPI and sustainability of our model.

5.3 Communication strategies

5.3.1 Types of data transfer

MPI supports communication types:

- *Point-to-point communication*. Data is sent from one node to another. Default communications are blocking: the send call blocks until the send buffer can be reclaimed. This implies that after the send, the sender can safely over-write the contents of a variable used for communication. The situation is similar on the receiving end: the

[3]MPI, or Message-Passing Interface, "is a message-passing application programmer interface, together with protocol and semantic specifications for how its features must behave in any implementation", which "includes point-to-point message passing and collective (global) operations, all scoped to a user-specified group of processes" Gropp et al. (1996).

receive function blocks until the receive buffer actually contains the contents of the message. If needed, MPI also supports non-blocking communications, which allow possible overlap of message transmission with computation, or of multiple message transmissions.

- *Collective communication*. This is used when information located on one cluster node must be shared with all the others, or when information scattered over the nodes must be gathered on one of them. A single instruction, called by all involved nodes, replaces a loop of point-to-point transmissions.

In the next subsection, naive implementations using these two types of data transfer are tested, in order to estimate the communication influence on simulation time.

5.3.2 Naive data transfer

5.3.2.1 What data is sent

As noted, spatial parallelisation based on lymph network structure means that communication is only required when agents are going from one node to another. In Chapter 4, we detailed the lymph node implementation. In particular, we highlighted that inter-node mobility is implemented through use of lists of "migrating" agents. During iterations of the parallel model, these lists are the only information that need be exchanged between cluster nodes. A *communication strategy* is, therefore, the decision basis for the frequency and the method of list transfer.

5.3.2.2 Point-to-point transmissions

A first strategy is to use point-to-point communication only, and to exchange data at the end of each iteration. From here on, we will refer to this strategy as "strategy 1", and improvements to this strategy will be referred as "1.x", (with $x \in$ [1,9]).
Strategy 1 implies point-to-point communications between each pair of nodes. First, each node must create $n - 1$ sublists (with n the number of nodes); each sublist contains the

agents going from the current node to a given other. Exchange of lists between the nodes involved can then take place. This can be implemented using the algorithm presented in Figure 5.1, and requires $n(n-1)$ list transfers.

```
For i = 1..(n-1)
   For j = (i+1)..n
      Exchange list of agents moving from i to j
      Exchange list of agents moving from j to i
   Endfor
Endfor
```

Figure 5.1: Principal algorithm for strategy 1

For MPI, each list transfer requires two communications, with one parameter of the MPI communication routines the size of the transmission. It means that, to send a list containing m values, we must first send m in a one-integer message, and then send the list. Strategy 1, therefore, leads to $2n(n-1)$ point-to-point communications.

5.3.2.3 Collective transmissions

Another strategy is to exclusively use collective communications. In the following, we refer to this as "strategy 2", and use the same naming convention as above for future improvements.

Here, each node will send the whole list to all the others. The first step is, therefore, the communication process, as presented in Figure 5.2. It represents n collective list transfers, i.e. $2n$ actual MPI collective communications. The next step is performed locally by each node: searching in all the received lists for the agents which are entering the node.

```
For i = 1..n
   Send list of agents leaving i to all other nodes
Endfor
```

Figure 5.2: Principal algorithm for strategy 2

5.3.2.4 Communication is indeed a bottle-neck

Strategies 1 and 2 were tested on a local cluster, for small configurations, (i.e. 4, 10, and 16 nodes), and short simulations. The results were discussed in Perrin et al. (2006c), and are here displayed in Table 5.1. The first remark is that communication is, indeed, a bottle-neck. Tested strategies are of course naive, and connectivity on the cluster is known not to be very recent, but these first results still highlight the need for an efficient communication strategy.

It is also important to note that local iteration time is largely constant and independent of communication strategies. This is a confirmation of the parallel nature of the problem.

Strategy 2 performs very poorly. This is due to the fact that, with this approach, destination nodes receive more data than they actually need, since they receive information about all the agents which left their host node, irrespective of their destination. There is a reduction in the number of communication steps[4], (n, compared to $n(n-1)$ with strategy 1), but this is obtained at too heavy a price: excessive data transfer eliminates strategy 2 as a viable solution.

Performances obtained from strategy 1 are more encouraging. They still highlight the need for further improvements, but communication overhead is more reasonable. A first solution is to consider the frequency at which lists are shared between nodes. More efficient communication implies sending non-empty lists. Obviously, the longer the interval between list transfers, the bigger this list gets, and the likelihood of sending an empty list decreases. In Table 5.2, computation times are shown for 20,000 iterations, when communication is performed at the end of every iteration or every other time-step. The program appears slightly faster when we communicate data less often. However the gain is not significant for very low agent count: in this case, few agents are scattered in the lymph node and are less likely to reach the exit point, even over several time-steps. The improvement is highest for medium agent count: for a high count, it is likely that at least one agent will reach the exit point during each iteration, and iterations leading to an empty list are, therefore, less com-

[4] and, therefore, also in the number of actual MPI communications

mon, but do occur. We observe an improvement when sending only at every other iteration; this pattern is confirmed if we wait three, four, or five iterations before sending the lists. There are, however, two limitations. The first is a memory concern, since an ever bigger list is resource-consuming. More importantly, there are biological considerations involved. A time-step is equivalent to fifty seconds, and the number of iterations must therefore be kept close to the actual time estimated for a cell to commute from one node to another. Separating the communication phases by more than five iterations is thus less realistic and should be avoided. Further improvements on the communication protocol are, therefore, required.

Cluster configuration		4 nodes	10 nodes	16 nodes
Strategy 1	Local iterations (s)	69	71	68
	Communication (s)	57	206	324
Strategy 2	Local iterations (s)	67	65	63
	Communication (s)	62	377	1239

Table 5.1: Naive strategies tested on small configurations

Communication frequency	Every iteration	Every other iteration
Low agent count	377 s	-1.48%
Medium agent count	982 s	-34.9%
High agent count	2187 s	-10.8%

Table 5.2: Influence of the list transfer frequency on communication time

5.3.3 Improving data transfer

In addition to eliminating solutions based on collective communication, early tests also provided a basis for improving strategy 1. In fact, direct transfer between every couple of nodes would on several occasions, due to MPI constraints detailed above, lead to sending information about an empty list, thus slowing the program down. For this approach to be efficient, we need a node to act as the intermediary between these pairs of nodes, with all the nodes sending their list to this one. On that node, the agents are sorted according to their

destination, and, to every node, a list is sent, containing only the agents which are relevant (strategy 1.1). This reduces the number of communication steps to $2(n-1)$. The main drawback for this improved communication protocol is that a node can only receive from (or send to) one other node at a time. It implies that in the meantime, the others are idle. Dedicating one node, (called node 0 hereafter), on the cluster solely to this role of intermediary ensures it is always ready to send and receive, rather than engaged in an iteration, and should reduce this shortcoming (strategy 1.2, see Figure 5.4a, p.74), though it will not eliminate it entirely.

Even if node 0 is available and immediately receives the list from node i, that node will be idle while waiting for list i of all agents migrating to node i, as this list can only be prepared once all incoming transmissions have been completed on node 0. Inclusion of an iteration between the sending of the first list, (agents leaving a node) and the reception of the second list, (agents arriving at that same node), prevents "computing nodes" from being idle, and gives time for node 0 to finish receiving every list and sorting the agents (strategy 1.3).

As the number of nodes increases, so does the time that a given node has to wait before being able to send/receive. An alternative is to create more "intermediary nodes", and subnetworks. For instance, on a 16-node cluster, we could have four groups, each formed with three "computing nodes" to deal with modelling and one node only used for communication. The first three would run an iteration, send their list, compute another iteration, and receive the new list. The other one receives the lists, shares information with other similar nodes, and sends the new lists. With this configuration, any node finds a maximum of three nodes in the queue at the time it joins, and the program is expected to be faster as a result (strategy 1.4, see Figure 5.4b, p.74).

To determine agent destination, the strategies, detailed so far, rely on a local function controlling inter-node cell mobility, (Chapter 4). The lymph network is not a "dense" network. Using terminology from graph theory, (see e.g. Diestel (2005)), it can be described as a directed finite graph which includes cycles, but is not a complete graph: if two lymph nodes are randomly chosen, it is likely that there will be no direct connections between them

(incomplete), even though there is always a path from one to the other (connected), (see Figure 5.3). These properties can be used to implement the lymph network. A communication network can be created explicitly, rather than only by a function as described above; communication can then be physically limited to this network (strategy 3). Without causing any biological issues, we can also impose the requirement that nodes have either two (one incoming and one outgoing) or three connections (two incoming and one outgoing, or vice versa). This would imply that for any given node, at any stage of the simulation, there is a maximum of two nodes in front of the queue.

A further improvement is to design this network to satisfy two-colouring, (strategy 3.1, see Figure 5.4c), thus decreasing the communication load: during odd iterations, black nodes send data and white ones receive it, and vice versa during even iterations.

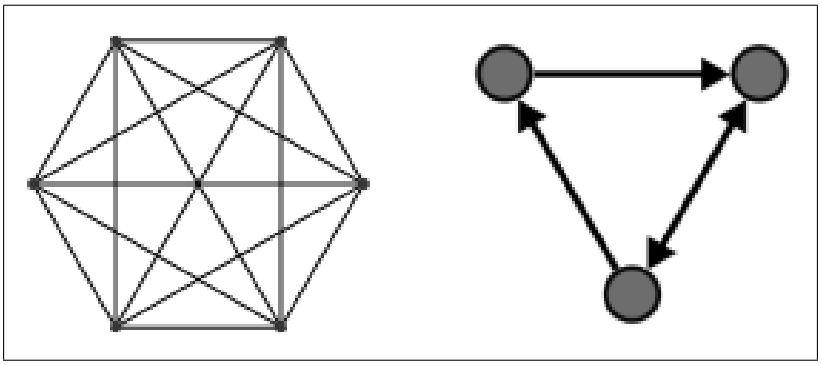
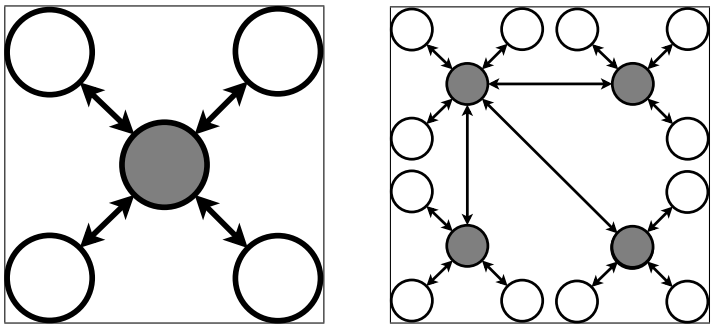


Figure 5.3: Complete graph with six nodes (left); directed and connected graph (right)

5.4 Validation

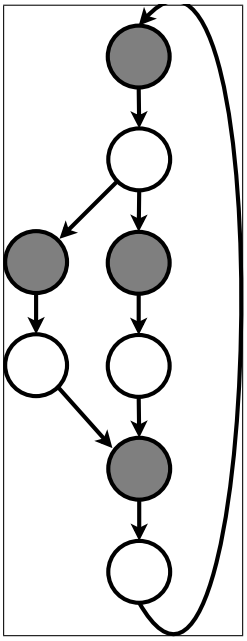
5.4.1 Final results on small clusters

All advanced strategies were tested on the same local cluster. In Table 5.3 (p.75), the results for 20,000 iterations and 16 nodes are shown. More advanced versions of the first strategy provide improved performance, with the notable exception of strategy 1.4. Tested on 16 nodes, it is faster than other implementations of strategy 1. We must mention, however, that



(a) Strategy 1.2

(b) Strategy 1.4



(c) Strategy 3.1

Figure 5.4: Communication strategies

this configuration means that only 12 nodes are used for actual biological simulation, while the other 4 nodes focus solely on data transfer. A fair comparison with other strategies must take this into account and look at results on smaller node counts for these. In that case, strategy 1.4 does not show any particular improvement, and was not, therefore, included in Table 5.3. Strategy 1.3, on the other hand, demonstrated useful performance and is a candidate for large-scale implementation.

Another promising candidate is the new strategy, in particular in its 3.1 coloured version. It shows significant improvements compared to any other communication protocol. The

fact that all 16 nodes are used for biological simulation is further encouragement: that is *this strategy combines efficient communication and better resource usage*. This can be explained by the number of communication steps; since each node has a maximum of three connexions, the number of transfers is of the order of $3n/2$, for a cluster of n nodes, (compared to $n(n-1)$ for strategy 1). Strategy 1.1. offers a count of the same order, $(2(n-1))$, but a greater idle time is observed. Indeed, with strategy 3.1, idle time for each node is reduced, as it is communicating with a small subset of nodes and, therefore, is less likely to wait for one already involved in another communication.

Communication strategy	Relative communication time
1	2.68
1.1	1.00
1.2	0.97
1.3	0.92
1.4	N/A
3	0.89
3.1	0.76

Table 5.3: Communication times for advanced strategies, compared to baseline strategy 1.1

5.4.2 Scaling-up

5.4.2.1 The need for a performance simulator

A model size of hundreds of nodes should lead to enhanced realism, but large clusters to handle this are a limited resource, and an optimal communication strategy is vital. This also implies that it is preferable to determine the optimal strategy before using these clusters. In other words, small clusters can be used for development, but larger clusters should be kept for actual biological simulations.

An interesting property of MPI is its very good scalability, so that an implementation validated on a small configuration has every chance of working on larger ones. This is still not enough in this case, however. What is required is an accurate estimate of communication time associated with each strategy, to satisfactorily assess which is the best one for

this particular model. As we saw with naive implementations 1 and 2, a communication strategy may have a critical model size above which performance is significantly reduced. This size was small for strategy 2, (which is only efficient on very small networks), but more advanced strategies may have a high critical point.
We considered that the best answer to this dilemma, (defining the best strategy without excessive physical testing), was to develop a performance simulator.

5.4.2.2 Evaluation of the strategies

The performance simulator, developed, has six parameters:

- Communication algorithm. This corresponds to the communication strategy currently tested, i.e. which node is sending to which node, type of communication, and time of the transmission.
- Node size. This is the size of the matrix modelling the lymph node.
- Agent count. Each node is initialised with a particular number of agents. This number is kept up-to-date throughout the performance simulation by taking into account the agents that are sent or received. For simplicity, population variations due to other biological variations, (e.g. production of virions by infected immune agents), are neglected in this context.
- Length of local iteration. This is a function of the node size and agent count.
- Network latency. This is a property of the hardware architecture under virtual evaluation.
- Bandwidth, or rate of data transfer. Another property of the simulated cluster.
- Length of data transfer. Transfer is obtained, (Equation 5.1, p.66), where data size is a probabilistic function of agent count.

This performance simulator was implemented, and validated using the results obtained on the local cluster, and strategies were then repeatedly tested. Of main interest here were performance measures, such as the average total execution time and communication time. Maximal values were also monitored, in case a particular configuration proved able to block a communication strategy or lead to abnormal performance. As shown in Table 5.4, tests focused on strategies 1.3 and 3.1, as these proved the most promising versions of the two types of communications. Again, as found for small configurations, strategy 3.1 performs best.

These results are consistent with expectations based on the configuration of communication steps. Strategy 1.3 leads to $n(n-1)$ communication steps, and strategy 3.1 to the order of $3n/2$, but the distribution of these data transmissions is very different. Strategy 1.3 puts an increasing load on node 0 as the number of nodes increases, while strategy 3.1 conserves the limit of three connections per node. Since the latter does not correspond to an increase in the size of transferred data, evolution of communication time is more satisfactory.

Importantly, this strategy was designed with biological realism in mind, in the sense that there is no unnecessary communication between nodes that are not connected in the lymph network structure. Increased efficiency of communications is not, therefore, balanced by any inaccuracy introduced into the model. As a consequence, this is the strategy we have chosen for large-scale model implementation.

Cluster configuration	16 nodes	32 nodes	64 nodes	128 nodes	256 nodes
Strategy 1.3	1.00	1.39	1.96	2.81	4.02
Strategy 3.1	0.83	0.89	0.96	1.05	1.17

Table 5.4: Performance evaluation on large clusters for strategies 1.3 and 3.1

5.5 Chapter summary

In this Chapter, it was demonstrated that a parallel implementation is both necessary and achievable. It is necessary, because of the size and complexity of the biological system to

be dealt with. Immune responses involve millions of cells, which are travelling through hundreds of lymph nodes. An accurate model of these complex interactions requires an important number of nodes and agents. This can not be implemented on a single computer long-term.

It is possible, however, because of the parallel nature of the biological system itself: apart from influence achieved through mobility of cells, lymph nodes act as *independent defense units*. This allows for efficient spatial parallelisation, by allocating each lymph node to each processor of a cluster-based configuration.

Even though such an implementation limits communication, designing an efficient communication strategy remains crucial, as highlighted through earlier tests with naive ones. Advanced strategies were, therefore, created, implemented with MPI, and tested on small cluster configurations.

Sensible use of computing resources also suggested the development of a performance simulator, used to evaluate communication strategies on larger configurations while saving these resources for actual biological simulations.

These tests allowed identification of *one strategy*, based on a *mimic of the lymph network*, as the *best communication protocol* for the immune system model.

This Chapter concludes, therefore, having built and tested an efficient parallel implementation of the lymph network, which is now suitable for large-scale, biologically-meaningful simulations. These are discussed in the next Chapter.

Chapter 6

Validation of the main model layer, and further improvements

6.1 Summary of current model

In Chapter 2, we detailed several existing immune models, identified the agent-based paradigm as promising, in the context of modelling immune response to HIV, and outlined limitations of current similar approaches, demonstrating a need for a new implementation.
Aspects of this model and implementation were detailed through Chapters 3 and 4, while selected features to exploit large computing resources, and the development of parallel solutions were discussed in the previous Chapter.
Challenges outlined as needing to be addressed, (Section 2.4.2), include overall modelling choices, (i.e. balance between agent diversity and agent population, explicit modelling of lymph nodes and reasonable granularity), as well as considerations of refined implementation, (e.g. cell-level implementation of immune memory, more refined antigen recognition).
Basic cell-level mechanisms have been validated, (Section 4.4.3), and the remaining tasks are to validate the other features implemented so far and, finally, to look at the inclusion of localised effects in the overall model.

6.2 Validation and results of the lymph network model

6.2.1 Computing efficiency of the lymph network model

Optimisation of the communication strategy was considered in Chapter 5, and permits simulations of large agent populations on configurations including several lymph nodes. Communication optimisation was performed on a cluster based on an outdated network hardware, which over-emphasised the importance of MPI data exchanges as a potential bottleneck. This allowed a better isolation of the communication component of the computing process and, in turn, the development of a very efficient protocol for data transfer between lymph nodes.

The focus here is on model performance when it is run on a more sophisticated and more recent cluster. Tests are performed on a 56-node cluster. Each "cluster node" has a dual-processor, and each processor has four cores. The cluster therefore offers 448 "computing cores" at 2.66 GHz, and modelling hundreds of lymph nodes is a reasonable target, (and would be on a scale similar to that of the whole immune system).

MPI allows specification of how many processes will be run on each "cluster node", (these are then scattered over cores of this node). For the lymph network model, one process is one lymph node. Of particular interest, therefore, is the evolution of the computation time:

- as a function of the number of agents at the start and the number of lymph nodes.
- as a function of the number of lymph nodes and the number of processes per "cluster node".

Table 6.1 displays the relative computation time as a function of the number of agents at the start, (a.p.n), and the number of lymph nodes, (l.n.), for simulations using eight processes per cluster node, (p.p.n.). Optimisation of local iteration allows simulation of very large populations of agents at a relatively low computing cost. Significant increases in agent count only generate moderate overheads in terms of computation time. This is very important, as immune response and viral spread both lead to variations of specific agent counts.

Table 6.2 displays the relative computation time as a function of the number of lymph nodes and the number of processes per cluster node, for simulations starting with 150,000 agents per node. As expected from performance simulations, optimisation of the communication strategy allows simulation of a large lymph network, of size similar to that of real system. This is crucial to model realism, as was confirmed by tests on cell mobility and its effects on immune activation and on viral propagation throughout an organism.

Implementation efforts, therefore, provide the opportunity for simulations reaching the scale of the immune system[1], both in terms of involved populations and "geographical" entities throughout the body. This is a significant increase in scale compared to existing models, and permits a more detailed representation of the immune system.

	8 l.n.	16 l.n.	32 l.n.	64 l.n.	128 l.n.	256 l.n.
∼30,000 a.p.n	1.00	1.04	1.08	1.12	1.16	1.21
∼150,000 a.p.n	1.08	1.10	1.12	1.16	1.24	1.29
∼300,000 a.p.n	1.21	1.22	1.23	1.24	1.27	1.31
∼600,000 a.p.n	1.58	1.58	1.59	1.60	1.64	1.78
∼1,500,000 a.p.n	1.89	1.90	1.92	1.93	2.11	2.11

Table 6.1: Model efficiency: relative computation time for several configurations of lymph nodes (l.n.) and agents per node at initialisation (a.p.n.)

	8 l.n.	16 l.n.	32 l.n.	64 l.n.	128 l.n.	256 l.n.	512 l.n.
1 p.p.n	1.00	1.06	1.14	N/A	N/A	N/A	N/A
2 p.p.n	1.08	1.09	1.10	1.14	N/A	N/A	N/A
4 p.p.n	1.25	1.26	1.27	1.28	1.30	N/A	N/A
8 p.p.n	1.68	1.71	1.75	1.81	1.94	2.00	N/A
16 p.p.n	N/A	3.47	3.52	3.54	3.57	3.58	4.04

Table 6.2: Model efficiency: relative computation time for several configurations of lymph nodes (l.n.) and processes per "cluster node" (p.p.n.)

[1] Using all 448 cores of the cluster available for testing, and allocating two lymph nodes to each, simulations involving more than one billion immune agents were successfully run.

6.2.2 Validating the biological features of the lymph network model

In validating a model of the lymph nodes and associated network, it is not possible to use cell counts seen in typical graphical representations of the three-phase disease progression, as these refer to peripheral blood. Moreover, cell counts from blood samples are known not to be an accurate measure of actual disease progression, as recirculating cell populations are significantly different from the overall cell populations. This means that a model, based on matching variations in peripheral blood cell counts, would, in fact, be unlikely to give a good representation of the overall immune response and disease progression through the lymph network. This approach is, however, common in existing models, which do not explicitly deal with lymph nodes, (see e.g. Castiglione et al. (2004)), and limits comparisons with our lymph network model.

Validation of our model must, therefore, be based on known signatures of HIV infection, (such as mutations and massive loss of memory cells), and evaluation of relevant model properties based on clinical data relevant to the lymph node context.

6.2.3 Balance between agent diversity and agent population

Abundance and diversity of cells are crucial to the efficiency of the immune response. The evolutionary argument is that, given the complexity and "cost" of maintaining such populations, this structure would not have been largely conserved between all vertebrates if it was not essential. More details on this "biological arms race" between invading organisms, (viruses, bacteria, fungi), and the host immune defenses can be found in the specialised literature, (e.g. Kasahara et al. (2004); Murphy et al. (2007)).

As complexity of a system increases, selection of the parameters and mechanisms to represent becomes less trivial. In the context of immune response, relatively simple cell-level interactions lead to considerable complexity of outcome for the system as a whole. Emergence of tissue-level and, even more crucially, body-level patterns are, nevertheless, clearly dependent on size of cell populations, as demonstrated e.g. by HIV progression: the whole immune system collapses once cell counts, (especially CD4 cells), decrease below critical

levels. Model realism therefore requires large agent populations and careful selection of parameters, given that accounting for every immune cell of every lineage is not a realistic target, even with current computing resources.

The lymph node model, therefore, implements three essential cell types, as well as viral agents, (Chapter 3). In the first instance, it guarantees enough diversity to account for most immune and viral mechanisms, and also permits simulation of large populations. The current model can, in fact, simulate up to two million agents per lymph node.

The parallel implementation of the lymph network model permits large-scale simulations, (Chapter 5). These simulations involve approximately 500 to 1000 lymph nodes, for a total population more than one billion agents. An adequate balance between agent diversity and large agent population is, therefore, obtained.

6.2.4 Explicit modelling of lymph nodes, and inter-node mobility

The lymph node model incorporates two levels of cell mobility. Intra-node mobility is crucial to physical contact between cells, and was validated, (Chapter 4). The chosen implementation for inter-node mobility was detailed, and the parallel structure for the lymph network model permits validation on large sets of lymph nodes.

Of particular interest in this context is to look at viral spread through the lymph network. Simulations are run on the 32-node lymph network shown in Figure 4.10, (p.61). In Table 6.3, (p.85), shown, for each node, is the iteration range of the first appearance of HIV, and the corresponding time since infection, with a precision of 10 iterations, (i.e. less than nine minutes), for the first simulated hundred minutes, and of 300 iterations, (i.e. less than five hours), subsequently. Due to cell mobility and the associated spread through the network, there are important variations of the time of infection for each individual node, from one simulation to another. Complete infection of the network, however, is always obtained after about a month, with a standard deviation of less than three days[2]. These results are in accordance with known patterns of viral spread through body. Information propagation in

[2] Average and standard deviation are obtained performing twenty simulations.

terms of immune activations follows a similar pattern, as could be expected.

The importance of cell mobility in the early stages of infection is clearly apparent, both from the delay of viral spread through the lymph network and the propagation of activated immune agents which initiate a body-wise response. Reproducing the progression provides an additional insight into the early stages of infection, and is a significant advance from previous models. In these models, even when the node and network structures are acknowledged as being essential, (see e.g. Baldazzi et al. (2006)), the implementation and the temporal granularity does not permit investigation on the node-to-node progression if the infection. This is, nevertheless, important, and may provide a basis for new treatments. The contribution of our model is, in that context, significant.

6.2.5 Immune memory

Transformation into memory cells is achieved by copying the agent activation state into the memory state, and restoring the former to initial state, (i.e. 0). Reactivation can then be obtained by physical contact with any viral or infected agent related to previous activation. Reactivated agents regain previous behaviour, but multiplication is faster, (Chapter 3).

To highlight the significance of this feature, tests are performed on a small, 16-node, lymph network, with a high viral load, (to obtain faster spread than that for Table 6.3), and with/without immune memory. A short infection is initiated, which is performed on a system including naive agents only, (i.e. there is no memory), and on a system which is initialised with 50% of agents created with memory of a previous infection, (for which the strain corresponding to that infection is randomly chosen). Results shown in Table 6.4, (p.87), represent the immune response to infection in both configurations during the first week after infection, in terms of time to detection of the infection and number of immune cells activated by HIV-related antigens and involved in the response in the twenty-four hours following this detection. The results shown correspond to two simulations initiated with the same seed for the random number generator, and the differences between the two are, therefore, a direct consequence of the inclusion of immune memory. These can be

	First local infection	Time since infection	Average	Standard dev.
Node 0	[27,600; 27,900]	16 days	11.9 days	6.1 days
Node 1	[31,200; 31,500]	18 days	13.2 days	6.2 days
Node 2	Origin of infection	0 minute	0 minute	0 minute
Node 3	[300; 600]	6 hours	64 hours	52 hours
Node 4	[2,700; 3000]	40 hours	96 hours	62 hours
Node 5	[16,500; 16,800]	10 days	7.7 days	4.7 days
Node 6	[18,300; 18,600]	11 days	11.2 days	7.0 days
Node 7	[23,100; 23,400]	13 days	10.4 days	6.4 days
Node 8	[34,800; 35,100]	20 days	4.3 days	5.1 days
Node 9	[37,500; 37,800]	22 days	5.9 days	5.0 days
Node 10	[39,600; 39,900]	23 days	15.1 days	8.3 days
Node 11	[44,400; 44,700]	26 days	16.4 days	8.1 days
Node 12	[46,800; 47,100]	27 days	17.9 days	7.9 days
Node 13	[49,200; 49,500]	28 days	18.7 days	7.8 days
Node 14	[57,300; 57,600]	33 days	22.6 days	9.6 days
Node 15	[49,800; 50,100]	29 days	24.2 days	11.4 days
Node 16	[44,400; 44,700]	26 days	24.9 days	10.2 days
Node 17	[46,500; 46,800]	27 days	25.7 days	9.9 days
Node 18	[49,800; 50,100]	29 days	27.4 days	13.5 days
Node 19	[51,300; 51, 600]	30 days	26.8 days	12.8 days
Node 20	[52,800; 53,100]	31 days	27.9 days	12.4 days
Node 21	[54,300; 54,600]	32 days	29.1 days	11.9 days
Node 22	[51,300; 51, 600]	30 days	29.3 days	12.4 days
Node 23	[31,200; 31,500]	18 days	16.9 days	8.9 days
Node 24	[32,400; 32,700]	19 days	17.6 days	7.6 days
Node 25	[35,100; 35,400]	20 days	18.9 days	7.7 days
Node 26	[36,600; 36,900]	21 days	20.1 days	7.7 days
Node 27	[39,900; 40,200]	23 days	21.4 days	7.6 days
Node 28	[43,500; 43,800]	25 days	22.7 days	7.5 days
Node 29	[46,500; 46,800]	27 days	16.1 days	7.2 days
Node 30	[49,800; 50,100]	29 days	17.4 days	7.3 days
Node 31	[50,100; 50,400]	29 days	26.1 days	8.2 days

Table 6.3: Inter-node mobility: influence on viral spread.
The delay between the initiation of the infection in the body and the first local infection in a lymph node is given, for a typical simulation, as a range, (for iterations), and as its equivalent in "real time". We also give the average time, and standard deviation.

observed, both in terms of rapidity and efficiency of immune response, when memory is taken into account. Another consequence is that, due to faster and more important activation, more potential targets are available for HIV infection and, as a result, viral infection also spreads faster and further during the first week. Here, only HIV-related patterns are reported, but the process is similar for other infections, with impact of immune memory on viral spread clearly crucial. As for previous tests reported in Table 6.3, variability can be observed, for a given node, from one simulation to another, but the overall pattern is conserved, and the inclusion of immune memory does not significantly alter the variability detailed in this earlier Table.

This is consistent with known mechanisms and offers satisfactory validation of this feature. The loss of memory cells is also observed in other models, such as Zhang et al. (2005). Our model, however, is a more accurate reflection of the effects of this loss. In Zhang et al. (2005), the immune memory is centrally controlled and, even though memory cells are similarly eliminated, the memory of past infections can not be lost. In our model each memory cell incorporates its small fraction of the overall memory, and the elimination of immune cells leads to a loss of memory: repeated infections are each treated as an initial infection, and the response becomes less efficient as the simulation progresses, which is a factor in the appearance of opportunistic diseases (Mathe et al., 1996).

6.2.6 Refined antigen recognition

In the lymph node model, antigen recognition is implemented so as to allow *adaptability*. This is obtained by using two lists to account for affinity between viral strains and immune clonotypes, (Chapter 3). Using a single list can be equivalent to distances commonly used in other models, (e.g. Hamming distance[3]). Addition of a second leads to more refined modelling, and inclusion of adaptability.

To demonstrate the significance of this second list, an experimental design similar to that for the immune memory tests is used: on a 16-node lymph network, simulations are performed

[3]The Hamming distance between two strings of equal length is the minimum number of substitutions required to change one into the other.

	No immune memory		With memory	
	First activation after local infection	HIV-activated population	First activation after local infection	HIV-activated population
Node 0	no activation	0%	infection first	$< 0.01\%$
Node 1	no activation	0%	infection first	$< 0.01\%$
Node 2	$\sim$1 min.	13.05%	$\sim$1 min.	21.72%
Node 3	$\sim$25 min.	1.05%	$\sim$5 min.	2.47%
Node 4	$<$ 4 h.	0.11%	$\sim$10 min.	0.41%
Node 5	infection first	$< 0.01\%$	infection first	0.09%
Node 6	no activation	0%	infection first	0.13%
Node 7	no activation	0%	no activation	0.24%
Node 8	no activation	0%	no activation	0.08%
Node 9	no activation	0%	no activation	$< 0.01\%$
Node 10	no activation	0%	no activation	0%
Node 11	no activation	0%	no activation	0%
Node 12	no activation	0%	infection first	0.03%
Node 13	no activation	0%	no activation	$< 0.01\%$
Node 14	infection first	$< 0.01\%$	infection first	0.11%
Node 15	no activation	0%	no activation	0.10%

Table 6.4: Immune memory: effects on rapidity and efficiency of response.
For both configurations we report, for the first week of infection, the delay in the spread of the immune response, (in "real time"). If the infection is reaching a node first, the response in that node may be a consequence of the infection, (i.e. local initiation of a new immune response), rather than due to the spread of the original immune response, and this is reported accordingly. When there is an immune response in a node, we also report the proportion of cells involved in this response, twenty-four hours after the start of the local response, (in "real time").

with and without second list. This is done for the first two weeks of infection, and, (as for memory tests), with a high viral load in order to obtain a faster spread than that reported in Table 6.3. Results are shown in Table 6.5.

As for Table 6.4, the results correspond to two simulations initiated with the same seed, to highlight the effect of the change of configuration. Clearly, adaptability leads to a more efficient immune response: viral infection was recognised more quickly in the node of origin, (node 2), and was better contained, as can seen from the increased activation delay in node 3, reached next. Adaptability also has a significant impact once the virus starts spreading from one lymph node to the next, and overall activation through the whole network is achieved faster. This influence is, however, a "double-edged sword", as increased

activation implies more infection targets. The damaging influence of chronic cell activation is a well-known component of overall dysregulation of the immune system associated with HIV infection (Brenchley et al., 2006; Munier and Kelleher, 2007; Tesselaar et al., 2002). The viral spread described in Table 6.3 is obtained from the complete model, which includes adaptability. The corresponding considerations on variability therefore also apply to the right column in Table 6.5. Using a single list, we observe a reduction of the standard deviation, (reduced by a third), which is a consequence of the limited immune activation.

	Using a single list	Including adaptability
Node 0	no activation	∼12 days
Node 1	no activation	no activation
Node 2	∼6 min.	∼2 min.
Node 3	∼22 min.	∼6 h.
Node 4	∼21 h.	∼12 h.
Node 5	∼10 days	∼29 h.
Node 6	∼9 days	∼21 h.
Node 7	∼10 days	∼12 days
Node 8	∼10 days	∼37 h.
Node 9	∼12 days	∼12 days
Node 10	no activation	∼13 days
Node 11	no activation	∼13 days
Node 12	∼11 days	∼12 days
Node 13	no activation	∼13 days
Node 14	no activation	∼13 days
Node 15	no activation	∼13 days

Table 6.5: Refined antigen recognition: effects on time of occurrence of first activations. We report the delay in local initiation of the immune response, in the first two weeks after infection. If during this period there is no HIV-related immune activity in a given lymph node, this is also indicated.

6.2.7 Long-term disease progression

As detailed in Sections 4.2 and 6.2.2, disease progression within each lymph node is significantly different from the well-known plots depicting the three-phase evolution of the disease, using blood samples. From these plots, even though the cell levels are not directly relevant to our study, the time scales are very useful, and can be used for validation.

Indeed, variations on blood samples, (shown in Figure 6.1, p.91), are the visible sign that interactions are taking place within lymph nodes. Sharp phase transitions in those samples indicate a change of regime in the lymph nodes.

To analyse our model, we focus on three of these critical points in disease progression: the peak and end of the initial acute phase, and the end of the latency period, which corresponds to AIDS onset.

The peak of acute infection corresponds, in the blood sample, to the point when viral load starts decreasing and the CD4 count increases. This is generally observed after six weeks. For our lymph node model, we consider this point reached when infection has spread through the whole lymph network and three quarters[4] of the nodes have a similar direction of change in cell counts.

The acute phase ends when, in the blood samples, viral load is back to low levels, and CD4 count starts decreasing again, (more very slowly). This typically occurs after approximately nine weeks. The main difference, in our context, is that viral load persists in lymph node, so we consider this point reach when, for three quarters of the node, the virion count is below 10% of its peak value and CD8 count is decreasing, (which is a sign that the viral load is not high enough to initiate a large-scale immune response in the corresponding lymph node).

The end of the latency period is characterised, from blood samples, by a significant increase of viral load and a CD4 count decreasing well below 300 cells/mm^3. Clinically, it corresponds to the onset of AIDS and the appearance of opportunistic diseases, which the immune system can no longer handle. Consequently, in our model, it corresponds to a sharp increase in viral load which is not followed by an increase of the CD8 count, (which would have signaled an immune response, even a weak one).

The results of our long-term simulations, based on these three points, are shown in Table 6.6, (p.91). The three phases are reproduced, and realistic orders are obtained for each time span, in particular the latency period. The length of the acute phase is slightly over-

[4]This is an arbitrary value, used to highlight the fact that we require a general trend. It should be noted, however, that variability between the nodes is, at this stage, more limited than in the very early stages of infection. Using other thresholds, (e.g. $2/3$ or $4/5$), does not significantly alter the results.

estimated. This may be due to recirculation dynamics or proportions of activated cells at the start of the run, which may need to be refined.

With respect to the length of the latency period, it is important to note that, while we obtain realistic values, (in terms of both average latency length and high variability between individual progressions), the model is not able to simulate long-term nonprogressors[5]. Such individuals are very rare, (less than 1% of patients), and it is not fully understood why they do not progress to AIDS. From the model results, it would seem the cause is related to an aspect not parametrised in our approach.

The model also reproduces some cases of early progression to AIDS, (as short as five years), but these do not qualify as rapid progressors[6]. Again, this may suggest for the cause for rapid progression is outside of what is taken into account in our current model.

Some parameters have an interesting impact on the length of the latency period. Increasing the mutation rate or the viral production of infected cells leads to sooner onset of AIDS, (as more viral strains are able to simultaneously target the immune system), while an increased list of immune clonotypes recognising each viral strain leads to a longer latency period, (unrealistically large lists would even lead to elimination of the virus, which is not surprising). The former is consistent with the current understanding of the infection dynamics, and the latter should be considered in the light of ongoing efforts to develop a vaccine to HIV.

In the context of possible future inclusion of medical treatment in our model, it is also important to note that our model gives a more detailed resolution on disease spread throughout the infection progression, compared to existing models. This is a consequence of our efforts on the early stages of the infection. The long-term progression is, in fact, a "sum" of successive short-term spreads: when a mutation occurs, the new strain contaminate the whole network and, once detected, is responsible for the initiation of an immune response. This process is similar to the initial infection. The magnitude and time scale of the response

[5]Long-term nonprogressors are individuals who have been living with HIV for over 10 years (there is no agreed time span, but authors generally use 10 to 12 years as a threshold), have stable $CD4^+$ counts of 600 or more cells per cubic millimeter of blood, show no sign of HIV-related diseases, and have not received any antiretroviral therapy.

[6]Rapid progressors are individuals who progress to AIDS within four years of HIV infection.

are different, (because the immune is already active, and may also be weakened), but the mechanisms involved, (such as cell mobility), are accurately handled by our model.

The end points we obtain are not different from previous models or clinical data, (which is not surprising, and is reassuring, since we are dealing with the same biological system), but the increased resolution of the simulation offers greater prospects for treatment inclusion, (see Sections 6.4.3 and 8.2).

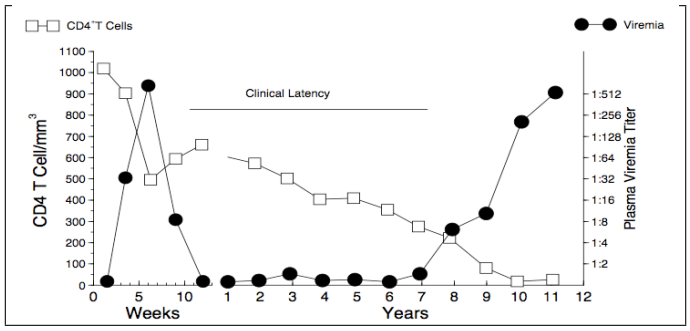


Figure 6.1: Standard three-phase disease progression, (reproduced from Zorzenon dos Santos and Coutinho (2001)).

	Average [Standard deviation]	Clinical data
Peak in acute phase	6.7 weeks [1.2]	∼6 weeks
End of acute phase	9.4 weeks [1.6]	∼9 weeks
End of latency period	8.0 years [3.7]	6-12 years

Table 6.6: Long-term disease progression. Comparison between clinical data, (not including rapid progressors and long-term nonprogressors), and time points obtained from the model.

6.3 Local and vital: early infection of the gastro-intestinal tract

Tests, detailed above, highlight the importance of cell mobility and explicit lymph node implementation, particularly for features of early infection. Viral spread from one lymph node to the next is not trivial, (with a crucial factor the internal state of agents within one node, characterised for instance by a higher proportion of activated CD4 cells when the spread of

viral infection is faster). This leads to formation of localised patterns.

When considering the whole immune system, a given area may exhibit very distinct patterns: this includes the gastrointestinal tract, (illustrated in Figure 6.2, p.93). Of major importance is the role of this tract in terms of immune population: it harbours the majority of the body's lymphocytes, where for instance blood only accounts for a few percent of these (Mehandru et al., 2005a).

Even more importantly, these cells are in close proximity to the external environment and are, therefore, constantly exposed to countless antigens, (food, microbes, etc.). This results in two crucial properties: more than 90% of these lymphocytes have a memory phenotype, and proportion of activated cells is significantly higher (Mehandru et al., 2005b; Schieferdecker et al., 1992). Another property, (high prevalence of CD4 cells expressing CCR5 receptor), is thought to be important, but quantification seems problematic, and apparent expression of CCR5 does not correlate with infection (Mattapallil et al., 2005).

These factors mean that there is, typically, a massive infection in the tract even in the early stages of HIV infection, and the response in that area is, therefore, also very active. Recently published experimental results show:

- A very rapid and very significant decline in $CD4^+$ counts, exceeding 25% after four weeks of infection (Guadalupe et al., 2003).
- Significant levels of infection and destruction observed even within days of infection for memory $CD4^+$ cells (Mattapallil et al., 2005).
- An increased cell proliferation in response to infection. The cell proliferation marker was found on 80% of intestinal $CD4^+$ cells four weeks after infection, as opposed to less than 10% in healthy patients (Guadalupe et al., 2003).

Due to this local but substantial depletion of immune cells, the overall cell population is also severely reduced, and this imposes significant pressure on the immune system in terms of memory pool maintenance (Brenchley et al., 2004). It also damages lymphoid tissue architecture, and this hinders the ability to support normal lymphocyte homeostasis and

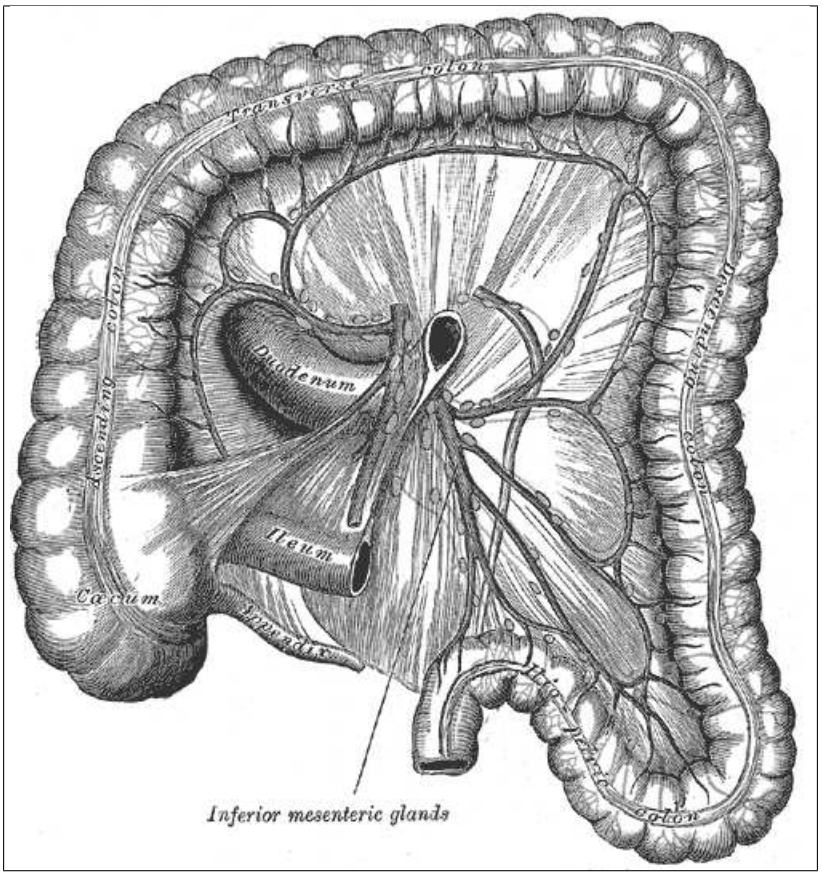


Figure 6.2: Some lymph nodes of the GI tract, (reproduced from Gray (1918))

antigen presentation.

Early infection in the gastrointestinal tract has, therefore, become an essential of research against HIV. Exact implications of GI tract infection remain largely unclear, but some interesting progress is being made. At the molecular level, it has been shown that preferential targeting of gut-associated $CD4^+$ cells may be due to interactions between viral glycoprotein gp120 and integrin $\alpha_4\beta_7$, which is specific to these cells, as it is required for migration through lymph nodes to *lamina propria*[7] of the gut-associated lymphoid tissue (GALT) (Sattentau, 2008).

At a higher level, restoration of cell populations after the acute phase is also under investi-

[7]The lamina propria mucosae, ("the mucosa's own special layer", in Latin), is a thin layer of tissue which, together with the epithelium, constitutes the mucosa. It is often referred to as *lamina propria*.

gation. Observations highlight a delayed and incomplete restoration of cell populations in chronically infected patients, even for those receiving highly active antiretroviral therapy, (HAART), for more than five years (Guadalupe et al., 2003). This standard therapy leads to restoration of cell levels in peripheral blood, but not in the tissue considered. A similar therapy, however, if initiated during primary infection, is effective in restoring cell populations. In this case, restoration is a consequence of cell recirculation and increased homing from the periphery of the tract, rather than local cell proliferation.

The extension of the current lymph network model aims to address the following questions:

- Is early infection in the gastrointestinal tract a good indicator of long-term progression throughout body?
- What is the effect of gastrointestinal tract infection on overall disease progression?
- Would efficient treatment of early gastrointestinal tract infection have a significantly beneficial effect on overall disease progression?

The remainder of this Chapter will detail early attempts to account for gastrointestinal tract infection and related effects.

6.4 Current implementation and future work

6.4.1 Accounting for localised properties

In an agent-based model, entities update their internal state based on interactions with other agents, but also with the environment itself. It is, therefore, possible to create *compartmental properties* which will locally alter agent behaviour.

Explicit implementation of the lymph nodes is crucial in the lymph network model. Since each node is modelled separately, it is possible to select a subset of them and add additional properties for agents in those nodes. For instance, a higher probability of activation by some foreign antigen, not related to HIV, can be specified.

Not all components of gut-associated lymphoid tissue are, in the strict sense, lymph nodes.

These also involve tonsils, Peyer's patches, or diffusely distributed lymphoid cells in the *lamina propria*. For most, however, function and structure is very similar to lymph nodes, and generic matrix structure used for node implementation can, therefore, also be used for these components.

Thus implementation of the gastrointestinal tract is obtained through selection of a long lymph chain in the lymph network structure and alteration of local properties:

- Selected nodes are initialised with agent populations reflecting known cell properties, e.g. 90% of memory cells, most of them active.
- Agent counts in these nodes are initialised to high values, so that these account for half of the overall agent population.
- The probability, for a non-active CD4 or CD8 agent, to be activated by a foreign antigen not related to HIV is increased, so as to maintain overall levels mentioned above.
- The input of "fresh" cells in those nodes is similarly altered.

6.4.2 Early results

To assess the proposed model extension, tests are first performed on a 24-node lymph network for which 10 lymph nodes are used to account for the gastrointestinal tract. These are indicated in Figure 6.3.

To estimate the influence of gastrointestinal tract infection on the overall disease progression, simulations are performed with and without local specification for nodes associated with the GI tract, (using, as previously, the same seed for random number generation). As shown in Table 6.7, (p.97), significant differences appear between the two "runs", and are due to GI tract inclusion[8]. In the simulation with no GI tract, as expected from previous results, the virus spreads rapidly through one lymph chain, but two weeks are not enough to

[8] As for previous tests on inclusion of new features, these two simulations are initialised with the same "seed", ensuring the differences are not due to stochastic variability.

obtain complete infection. In the second simulation, however, once the GI tract is infected, the virus finds large amounts of potential targets, and viral spread is enhanced. After two weeks, most of the network is infected. Another difference is in the proportion of infected cells: ten days after local infection, twice as many cells are infected in node 11 during the second simulation, amounting to 70% of overall node cell count. This value is in agreement with biological studies, which found peak infection occured at day 10-11, with a subsequent loss of 60 to 80% of memory cells (Mattapallil et al., 2005).

GI tract inclusion clearly affects the overall viral spread, but does not appear to alter the variability reported in Table 6.3. This is due to the fact that this variability is a consequence of inter-node cell mobility, which is not altered by this new model feature.

Similar patterns are observed for simulations on larger networks. Thus the proposed model successfully implements gastrointestinal tract properties. Since such an attempt has not been previously reported, and given the importance of the tract in early disease progression, this is a useful complement to existing models.

It is also important to consider the effects of this new model feature on long-term progression, as reported in Table 6.8, (p.99). Including the GI tract reduces the acute phase, which is expected for the short-term simulations. The model results are now realistically reproducing observed time scales. The variability of this phase is not significantly altered, and the small difference is consistent with the current knowledge on the influence of the GI tract: because of the reaction of the tract to the infection, the overall peak is in part determined by the time the infection reaches the tract, (with a local peak ten days after local infection, as detailed above), which increases variability compared to a “uniform” network. Conversely, since cell depletion is increased in the GI tract, the acute phase is locally shorter, and often ends before the overall acute phase, therefore slightly reducing the overall variability of the latter.

With respect to the length of the latency period, obtained values are slightly lower than clinical results, but the difference is not significant. Overall, GI tract inclusion therefore improves the model behaviour. This is confirmed by looking at ratio between the peak of

acute phase, (t), and the end of the latency period, (T). The ratio between these two time points, (which are the easiest to observe and quantify), is just under 70, (~ 69.6). In the initial model, we obtained $T/t = 62.3$ while, with the GI tract included, $T/t = 66.7$. Refining this early GI tract implementation will probably further improve model realism, (and, therefore, this ratio). Our understanding of the model is also that the function used to simulate the input of new cells may now be the limiting factor, and that explicit thymus implementation may significantly enhance the model realism.

	Standard network	Including GI tract
Node 0	$\sim$ 6 days	$\sim$ 4 days
Node 1	$\sim$ 7 days	$\sim$ 6 days
Node 2	origin of infection	origin of infection
Node 3	$\sim$ 6 min.	$\sim$ 2 h.
Node 4	$\sim$ 10 h.	$\sim$ 17 h.
Node 5	$\sim$ 42 h.	$\sim$ 29 h.
Node 6	$\sim$ 3 days	$\sim$ 46 h.
Node 7	$\sim$ 5 days	$\sim$ 4 days
Node 8	$\sim$ 5 days	$\sim$ 3 days
Node 9	$\sim$ 6 days	$\sim$ 5 days
Node 10	$\sim$ 7 days	$\sim$ 6 days
Node 11	$\sim$ 10 h.	$\sim$ 8 h.
Node 12	$\sim$ 17 h.	$\sim$ 29 h.
Node 13	no infection	$\sim$ 5 days
Node 14	no infection	$\sim$ 7 days
Node 15	$\sim$ 6 days	$\sim$ 5 days
Node 16	$\sim$ 7 days	$\sim$ 6 days
Node 17	no infection	$\sim$ 10 days
Node 18	no infection	$\sim$ 11 days
Node 19	no infection	$\sim$ 12 days
Node 20	no infection	no infection
Node 21	no infection	$\sim$ 12 days
Node 22	no infection	$\sim$ 8 days
Node 23	$\sim$ 8 days	$\sim$ 7 days

Table 6.7: Gastrointestinal tract: effects on disease progression during first two weeks of infection (time of first local infection)

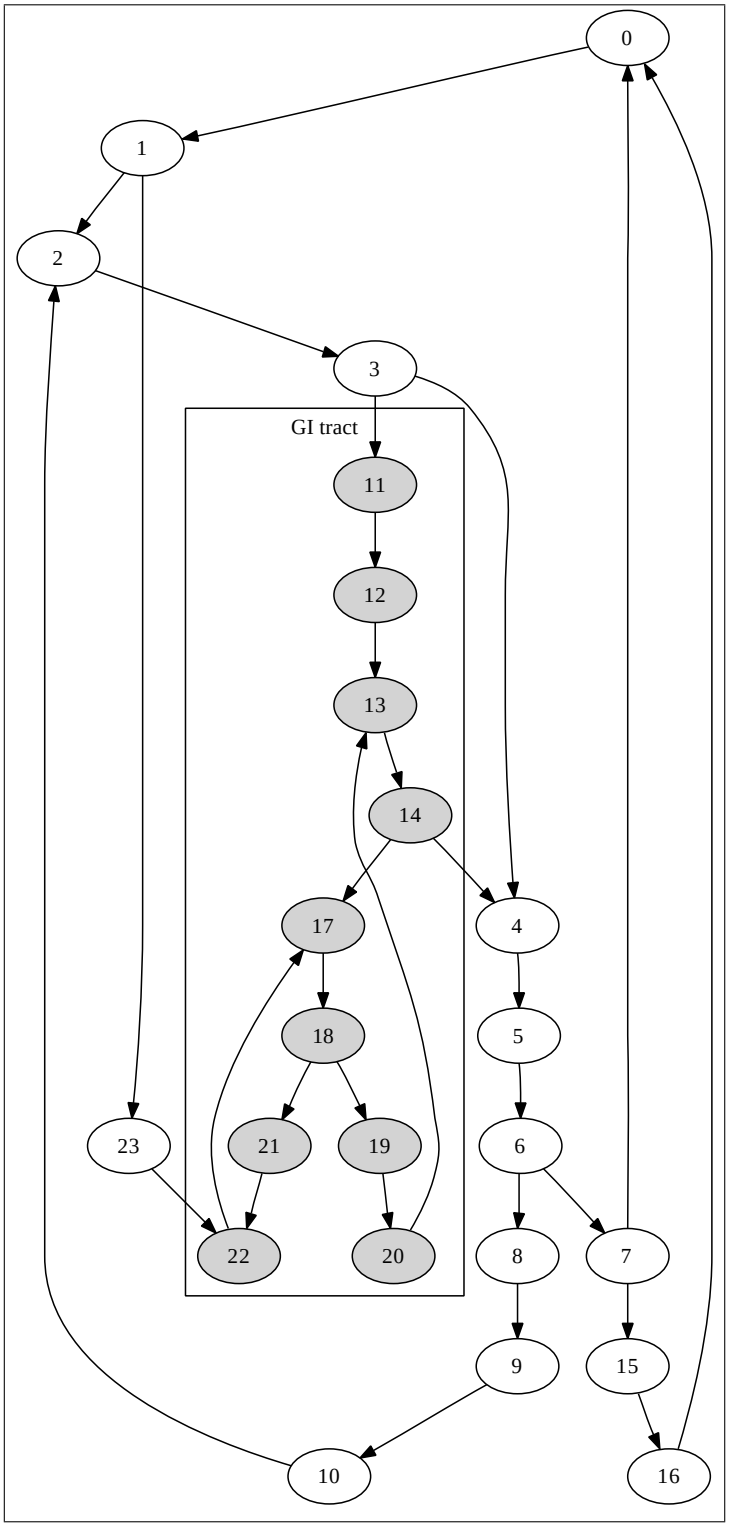


Figure 6.3: Lymph network used for tests on GI tract, represented by grey nodes

	Standard network	Including GI tract
Peak in acute phase	6.7 weeks [1.2]	6.1 weeks [1.4]
End of acute phase	9.4 weeks [1.6]	8.9 weeks [1.4]
End of latency period	8.0 years [3.7]	7.8 years [3.7]

Table 6.8: Long-term disease progression. Comparison between clinical data, (not including rapid progressors and long-term nonprogressors), and time points obtained from the model.

6.4.3 Future work

Initial tests on the gastrointestinal tract extension are very promising and confirm explicit lymph node implementation a useful basis for model development. Further simulations will provide valuable insight into importance of the gastrointestinal tract on overall disease progression.

In particular, further tests will be aimed at clarifying whether early infection in the GI tract can be seen as a prototype of overall infection patterns over the whole course of disease progression. To this end, it is necessary to look more closely at some signatures of HIV infection, and in particular viral mutation. Latency is characterised by an increasing number of viral strains, each maintaining a relatively small population: when one becomes too prominent, probability of successful recognition by the immune system increases, and it is soon eradicated.

A future focus will also be on assessing suitability of *early treatment* in the gastrointestinal tract as a way to better contain disease progression. Early indications are that this could be an interesting prospect for drug development (Guadalupe et al., 2003), but *in silico* investigation could provide further understanding of dynamics of cell population restoration. To this end, accurate modelling of treatment is required, so that different intervention frameworks can be evaluated.

This extension, (treatment modelling), requires detailed consideration. Could it be included as additional constraints on agent behaviour, or does it demand implementation of a new agent type? Drug dissolution and interactions between drug, immune system and virus are

non-trivial. Expertise in modelling these phenomena, however, exists within our research group: results from drug dissolution simulations, (see e.g. Barat et al. (2006a;b)), may, in the medium term, be adapted to account for diffusion of anti-viral treatment over the lymph network, and models of bacteria-antibiotics interactions, (see e.g. Walshe (2006)), could provide an interesting basis for analysis of virus-drug complexes.

6.5 Chapter summary

In this Chapter, modelling choices, detailed earlier, were successfully combined to facilitate emergence of known macroscopic signatures of HIV infection.
Cell mobility, in particular, results in a realistic representation of the viral spread through the lymph network. This is a consequence of detailed implementation of this particular feature, with significant efforts to improve local update optimisation and overall communication strategy.
These efforts permit simulation of hundreds of lymph nodes and more than a billion agents. This is a scale similar to that of the real immune system, and provides a useful insight into how observed patterns emerge from very precise local interactions.
All limitations identified in existing models have now been addressed:

- The balance between agent diversity and agent population is adequate, since simulations can involve a very large number of agents, (cell population several hundred times larger than in existing models), from types which are sufficient for the obtention of known immune and viral characteristics.

- Explicit modelling of lymph nodes has been implemented and validated, and proves a very important feature.

- A reasonable granularity, to account for cell-level interactions and movements, is guaranteed, as code optimisation allows time step of one minute.

- Cell-level implementation of immune memory is successful, and observed results are

in agreement with biomedical studies.

- More refined antigen recognition provides adaptability to the immune system, and effects on overall behaviour have been successfully assessed.

- The inclusion of localised effects is under way, and early results confirm the suitability of the lymph network model for such an objective.

This Chapter concludes, therefore, with a large-scale agent-based model which significantly advances the field of immune modelling. This model is based on a bottom-up approach and, as a consequence, further improvements will the result of a refined agent implementation. The next Chapter will, therefore, introduce methods to further improve model realism: to better understand the immune system, it is crucial to consider how this system is obtained. This requires additional layers to the current overall model, at the genetic and epigenetic levels.

Chapter 7

Beyond genotype: epigenetic modelling

7.1 Phenotypical immune response as a consequence of gene expression

7.1.1 Motivations

In previous Chapters, we saw how, using a bottom-up approach, based on cell-level interactions and using a large-scale implementation of the lymph network, it is possible to gain a better insight into the dynamics of disease progression. If agent implementation can be refined, overall model realism will be enhanced.

By definition, the immune system is a phenotypical system: we have considered and implemented "visible" cell characteristics. A better understanding of the development of these would, in turn, improve the accuracy of the modelled interactions and, therefore, of the overall immune system model behaviour.

In this context, it is useful to look at another layer of the immune system, and study how the immune responses patterns are obtained. In particular, how do various gene expression patterns result in different phenotypical expression? In other words, what are the mecha-

nisms for acquisition of phenotypical characteristics?

To analyse gene expression and function, microarray experiments are commonly used, and specific datasets dedicated to the immune system are appearing, (even though they are not currently available for humans). To permit analysis of such datasets, we developed a bi-clustering technique, (see Appendix A for details).

However, as will be detailed next, the phenotype is not a deterministic consequence of the genotype, and is controlled by *epigenetic* mechanisms. A further layer is, therefore, added to the model, to permit a more detailed investigation into these changes, both in the context of the main model layer and for other biological systems, (e.g. cancer initiation). The connection between the model layers is shown in Figure 7.1, (p.104).

In this Chapter, we introduce our objectives for this layer and present the current status of research on Epigenetics, before detailing a model of infection-induced epigenetic changes.

7.1.2 Gene expression is controlled by epigenetic changes

Early advances in Genetics led to the *all-genetic* paradigm: the phenotype is a deterministic consequence of the genotype. Obvious counter-examples were outlined and this was later amended and expressed using $P = G + E$, encompassing the notion that the visible characteristics of a living organism (i.e. the phenotype, P) combine hereditary genetic (i.e. the genotype, G) and environmental factors (E).

However, this formula fails to explain cell differentiation from a zygote, and why, for high heritability diseases such as schizophrenia, differences between monozygotic twins can be seen. Furthermore, identification of environmental factors, (e.g. smoking Mucha et al. (2006) and air quality Spurny (1996) for lung cancer), is relatively rare.

Early work Waddington (1949), and more recently over the last decade, (see e.g. Bird (2002); Wilkins (2005), has emphasised that genotype expression can be altered without changing DNA sequence itself, and tagged this phenomenon as *Epigenetics*: $P = G + E + EpiG$.

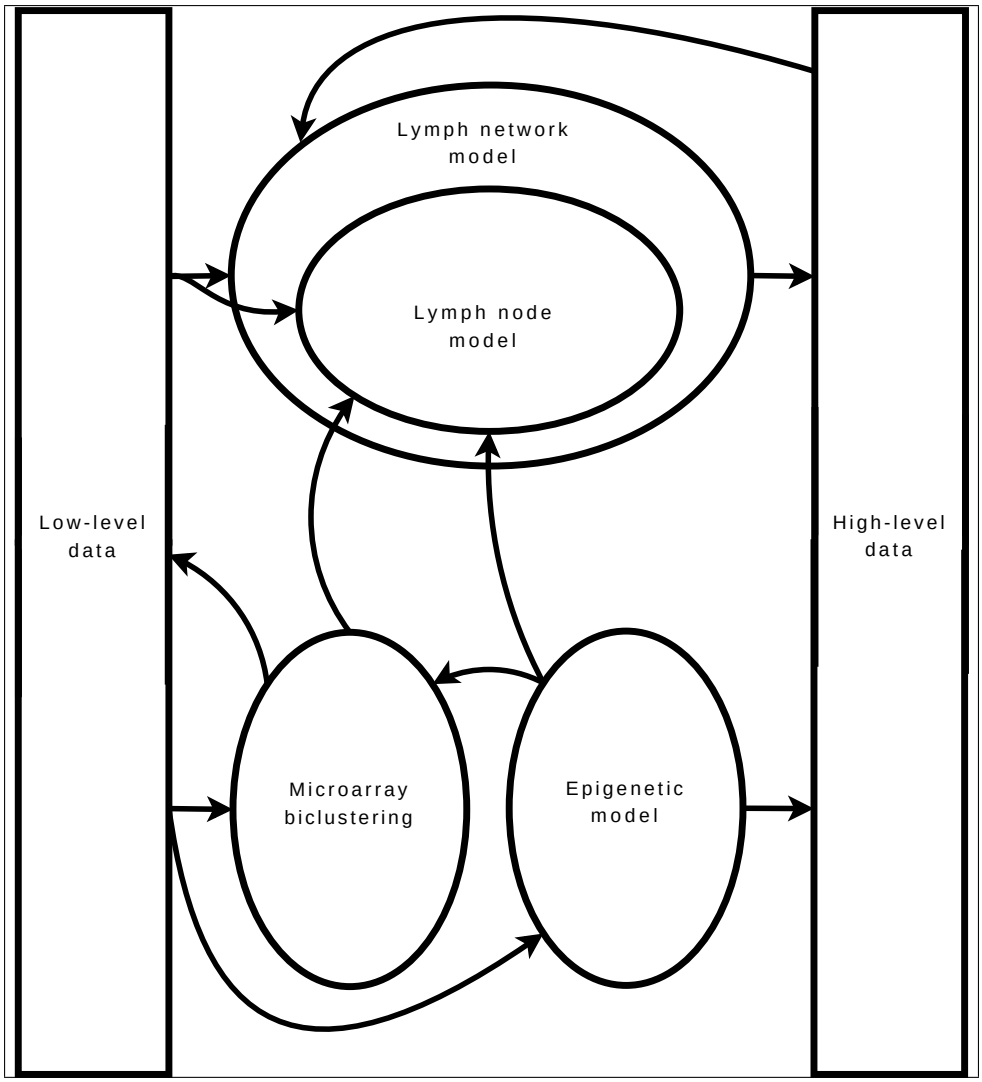


Figure 7.1: Epigenetic modelling as an additional layer of the overall model.
Low-level biomedical information, (e.g. genetic and epigenetic background) is used as an input to this layer. The results from the model are obtained at a low level, (i.e. epigenetic status, gene expression), and at a high-level, (e.g. effects of these mechanisms on disease development). They are useful in every research field where epigenetic changes are involved, including immune modelling, as they can be used for instance to refine the agent implementation in the lymph node model.

Epigenetic mechanisms involve heritable alterations in *chromatin structure*, (e.g. *DNA methylation* and *histone acetylation*, detailed in Section 7.2), amongst other epigenetic "signatures". In turn these regulate gene expression, but do not involve changes in DNA sequence. These "stable alterations" arise during development and cell proliferation and persist through cell division. While information within the genetic material is not changed, instructions for its assembly and interpretation may be.

7.1.3 Objectives

Similarly to the immune system, epigenetic mechanisms form a very complex biological system, and the objective here is to propose early models of such phenomena. Results obtained, in the long term, from these models can then be used to refine the lymph node model, as well as to better describe systems such cancer initiation or neural development.

Bioinformatics methods have been applied to Epigenetics, for instance to predict methylation status of a given DNA sequence (Das et al., 2006), but Computational Biology has yet to contribute in a significant way. In particular, no previous modelling work was found to exist on epigenetic changes, and this Chapter provides a stepping stone towards a large-scale project, (which we recently detailed (Perrin et al., 2008)). An analogy can be drawn with early modelling work on the immune response to HIV. Research on Epigenetics is in a position similar to that of research on HIV over a decade ago: the basics of the infection were understood, but lab testing was difficult, and *in vivo* testing implied evident ethical issues, which meant that quantitative data were sparse. Models developed at that stage were able to match some signatures of disease progression, (as outlined in Chapter 2), and were, therefore, used as a *proof of concept* for computational immunology. Ongoing refinement of proposed approaches has led to recent models, including that which has been proposed in this Thesis.

From collaboration discussions with the National Cancer Center (Tokyo, Japan), the chosen focus for an early model is aberrant DNA methylation induced by H. Pylori infection.

7.2 Epigenetic changes, interactions and perturbations

7.2.1 Chromatin structures

Chromosomes, which store all genomic information, are formed by a complex combination, called chromatin, of DNA and proteins. The major proteins involved are histones. Nine histones combine to form a nucleosome, shown in Figure 7.2. The characteristic structure of a nucleosome is that of four pairs of histones forming a core around which about 146 base pairs of DNA are wrapped. This is maintained in place by a linker histone, H1, and repeats over the chromatin every 200 base pairs. The remaining 50 base pairs of this repeating unit consist of "linker DNA".

This structure is crucial in gene expression: when a gene needs be expressed, several proteins must interact with it and nucleosome are, therefore, far apart, to facilitate access. In contrast, condensation of chromatin structure leads to gene silencing. Several epigenetic changes on histone structures participate in the control of these dynamics.

The link between nucleosome positioning and epigenetic changes is two-directional, and the positioning influences DNA methylation. Linker DNA is very susceptible to methylation changes, while core DNA is very difficult to methylate. As an additional complication, nucleosome positioning dynamically evolves, and is both DNA-dependent and energy-dependent. Changes in DNA methylation can also lead to nucleosome repositioning. Until recently, dynamics of these position changes *in vivo* were poorly understood, but new biological techniques are now being developed and offer promising results, (see e.g. Davey et al. (2003); Fatemi et al. (2005); Pennings et al. (2005); Segal et al. (2006)).

7.2.2 DNA methylation

DNA methylation corresponds to addition of a methyl group to a DNA strand. In humans, only 1% of DNA bases undergo DNA methylation. In differentiated cells, DNA methylation is typically limited to CpG dinucleotides[1]. Non-CpG methylation can be found in

[1] CpG: a cytosine followed by a guanine in the DNA sequence.

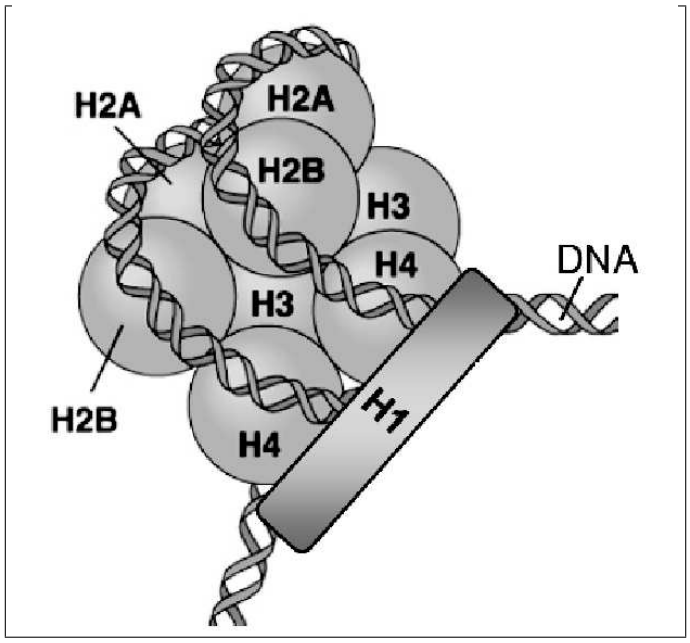


Figure 7.2: Schematic representation of a nucleosome, (adapted from Brenner (2005))

embryonic stem cells (Dodge et al., 2002).

Of particular interest are *CpG islands*. These correspond to areas with higher proportion of CpG, and are formally defined as follows (Gardiner-Garden and Frommer, 1987):

- Length of the considered region is at least 200 base pairs.
- GC percentage is greater than 50%, (i.e. more than half of amino-acids are cytosine or guanine).
- Observed/expected CpG ratio that is greater than 60%.

In humans, these islands are found in or near to 70% of gene promoters (Saxonov et al., 2006). While most CpG are methylated over the genome, these regions have a very distinct pattern: methylation of a CpG island corresponds to silencing of the associated gene. Aberrant changes in CpG island methylation are, therefore, linked with abnormal gene expression.

7.2.3 Histone modifications

Histone modifications correspond to the addition, (or removal), of a functional group, (methyl, acetyl, etc.), to specific amino acids of histone proteins. As detailed above, these proteins form a nucleosome core, around which DNA strands wrap. For each nucleosome, nine histones are required: two of each class H2A, H2B, H3 and H4, and one H1.

Modifications can occur on tails of histones H3 and H4, and in the core of H2A and H3. Some amino-acids can undergo several successive modifications. Lysine 79 of histone H3 can, for instance, be mono-, di-, or trimethylated (Barski et al., 2007).

Role of changes is modification-specific. For instance, trimethylation of H3K9[2] is associated with gene silencing (Barski et al., 2007), while acetylation of the same amino acid is linked with gene activation (Koch et al., 2007). It is also molecule-specific, as effects of trimethylation of H3K9 and H3K4 have opposite consequences on gene expression (Barski et al., 2007; Koch et al., 2007).

7.2.4 Interactions

While other epigenetic mechanisms exist, such as perturbations by siRNA, (small interfering RNA[3]), and piRNA, (Piwi-interacting RNA), they will not be detailed here. Lack of detailed information on these is even more limiting than for DNA methylation or histone modifications, and they can not, therefore, be an immediate target for model development.

Those two mechanisms are, however, sufficient to explore the behaviour of epigenetic systems. A complex interaction exists between DNA methylation and histone modifications, and epigenetic changes have different dynamics and stability. For instance, histone deacetylation is considered very rapid, while histone methylation is slow and DNA methylation very stable. Each modification is due to a specific family of enzymes[4]. These will not be detailed here, but it is important to note that such enzymes are part of larger complexes,

[2] H3K9: Lysine 9 of histone H3.

[3] Also known as short interfering RNA, or silencing RNA.

[4] e.g. DNA methyltransferases Dnmt1, Dnmt3a and Dnmt3b are controlling DNA methylation in mammals.

which lead to chain reactions. For instance, MeCP2[5] binds to methylated DNA and then forms a complex with HDAC[6], leading to histone modifications, (including methylation, through later recruitment of another complex involving Polycomb-group proteins). This is crucial during cell division. DNA methylation is, indeed, the only epigenetic change directly conserved during cell division, but through recruited MeCP2-HDAC complex and further interactions, histone modifications are restored.

DNA methylation can, therefore, be seen as the "lock" of gene silencing, (since it leads to recruitment of complexes reinforcing unstable histone modifications), but also as its "memory", (since its conservation during cell division ensures restoration of associated changes)[7].

There are, in addition to this, complex interactions within histone-related mechanisms. These are collectively known as the *histone code* (Gilmore and Washburn, 2007; Jenuwein and Allis, 2001; Strahl and Allis, 2000). An exhaustive list of all these interactions will not be provided here but, for instance, phosphorylation of serine H3S10 facilitate acetylation of H3K14 and inhibits methylation of H3K9, in turn enhancing gene transcription, while simultaneous acetylation of H4K5, H4K8, H4K12 and H4K16 inhibits methylation of arginine H4R3.

7.2.5 Perturbation of epigenetic patterns

Epigenetic mechanisms are involved in normal cell differentiation. All cells of an organism have, indeed, the same genomic information, and epigenetic changes are part of the process which switches genes on or off and leads to cell differentiation. Interactions between epigenetic changes in the early stages of development are, therefore, a crucial topic for stem cell research, (see e.g. Nimura et al. (2001)).

If epigenetic patterns are involved in normal gene silencing as part of cell differentiation, aberrant changes can also lead to abnormal silencing or transcription.

Alterations in DNA methylation, imprinting and chromatin structure are common in cancer

[5] MeCP2: methyl CpG binding protein 2.

[6] HDAC: histone deacetylase.

[7] This representation was proposed by Prof. Shoji Tajima when I visited his Laboratory of Epigenetics, at the Institute for Protein Research (Osaka University, Japan).

and links to epigenetic changes have been established in several cases, e.g. in Wilm's tumour (Rainier et al., 1993) and colon cancer (Liou et al., 2007), both of which involve Loss of Imprinting, (LOI, silencing specific genes on a parent lineage). In the latter, apparently predating the tumours, LOI occurs in surrounding tissue - i.e. an environmental or "field effect". Motivation for this type of research is that, if information is accessible, this may yeld pre-critical information on tumour formation.

Epigenetic mechanisms are also studied in other medical fields because of association with obesity (Cooney et al., 2002), abnormal neural development (Kubota, 2008), mood disorders such as stress vulnerability and bipolar disorder (McGowan and Kato, 2008), or risk of heart failure (Mano, 2008).

Common to this research is costly and time-consuming lab testing. Ethical issues also arise (study of epigenetically induced differentiation of stem cells being an obvious example). Another limitation is that, while successful in investigating specific phenomena, they so far fail to explain system-wide complex interactions. This can be explained both by overall system complexity and by technical constraints leading most research groups to focus on one epigenetic change in one given context.

This need for integration of these partial results is crucial to understanding the overall biological system, and computer-based modelling, (increasingly used as a complement to lab testing in other fields, see e.g. Dove (2006)), can provide useful framework to address such need.

7.3 Immune response and epigenetic changes

7.3.1 Epigenetics in the immune system

Agent-based models and other bottom-up approaches are often used to examine individuality in a system. In the lymph network model implemented, individuality of response is obtained through the implementation of local rules. At the epigenetic level, another layer of individuality can be observed: cells, through differentiation, have individual profiles.

Epigenetic changes are involved in differentiation of CD4 cells. Naive CD4 T cells develop into either type Th1 or type Th2 CD4 cells that predominantly secrete IFN-γ or IL-4, respectively. The former are mainly interacting with macrophages and cytotoxic $CD8^+$ T cells, promoting cell-mediated response. The latter interact with B lymphocytes and promote humoral response. This differentiation is obtained by the expression of one cytokine gene and the permanent silencing of the other, controlled using epigenetic mechanisms. Recent research, which focuses on isolating environmentally induced epigenetic change that occurs during Th1/Th2 cell development, could explain how certain Th1/Th2-associated conditions develop (Sanders, 2006). The authors hypothesize that diet, ageing, or use of certain drugs could lead to changes responsible for shift in Th1/Th2 profile which would, in turn, affect disease susceptibility and resistance.

In a recent article (Chang and Aune, 2007), the authors also considered those two lineages, and focused on the locus encoding interferon-γ (*Ifng* locus). In particular, they explored histone modifications. Methylation of H3K9 across the locus was found to be rapidly induced during differentiation, and to be conserved in Th1 cells. On the contrary, for Th2 cells, methylation is limited to H3K27. With much of the immune experience dependent on initial priming (Ruskin and Burns, 2006), it is evident that understanding epigenetic changes can provide further information on initial system status and differentiation of immune cells.

7.3.2 HIV-related epigenetic changes

Epigenetic patterns, related to HIV infection, are also under consideration. There is a growing argument that highly active antiretroviral therapy is not a viable solution to stop disease progression, as multi-drug resistant HIV (MDR-HIV) strains have appeared (Little et al., 2002). Alternative therapies are currently being sought, and epigenetic inhibitors such as peptide nucleic acids (PNA), which targets transcription of specific regions of viral mRNA, are considered to be good prospects (Sei, 2005).

Research is also focusing on methylation patterns. Methylation of specific regions of viral genetic material has been shown to be associated with inhibition of transcription (Bednarik

et al., 1987; Schulze-Forster et al., 1990), which implies, potentially, an important role in viral latency. Recent studies have confirmed this, showing that CpG sites in the 5' long terminal repeat (LTR[8]) are selectively hypermethylated, and that TNF-α-induced reactivation is associated with demethylation of the 5' LTR (Ishida et al., 2006). *In vivo* experiments are, however, limited by low copy numbers of HIV provirus, which prevent direct analysis of CpG methylation.

Further studies are, therefore, required to fully understand involved mechanisms, but epigenetic "treatment" is a promising long-term project. This, of course, implies a better comprehension of multiple epigenetic interactions, and *in silico* models would certainly advance this.

7.4 An example of infection-induced epigenetic perturbation

7.4.1 Context of the study

As a *proof of concept* for computational models of epigenetic changes, a study on infection-induced epigenetic perturbations is presented here. The model and target medical condition are the result of initial discussions on collaboration with Toshikazu Ushijima[9] and his team. Their main objective is to better understand epigenetic changes in the context of cancer initiation, and to apply this to improve prevention, early-stage diagnostics and treatment.

The motivation is that epigenetic alterations in non-disease tissues can be used as markers for disease risk and past exposure to some disease-inducing factors. In particular, current focus in on detection of aberrant DNA methylation in non-cancerous gastric mucosae, as the presence of such patterns can be used as a marker for both the risk of gastric cancers and past exposure to *Helicobacter pylori*. A detailed presentation in the context of this study is available in Nakajima et al. (2008). For convenience, main points are summarised here.

[8]LTRs are characteristic of viral genetic material. They flank functional genes, and their main function is to mediate integration of the retroviral DNA into host chromosome. The five prime, (5'), end refers to the end of the DNA, (or RNA), strand that has the fifth carbon in the sugar-ring of the (deoxy)ribose at its terminus, (as opposed to the 3' end, which is terminating at the hydroxyl (-OH) group of the third carbon in the sugar-ring, and is also known as the *tail end*.

[9]Carcinogenesis Division, National Cancer Center Research Institute, Tokyo, Japan.

In cancer development, aberrant DNA methylation is involved at two levels:

- Overall *hypomethylation*, which affects repetitive DNA sequences and causes both chromosomal instability, (and, therefore, tumours (Gaudet et al., 2003)), and aberrant expression of normally methylated genes (Smet et al., 1999).

- Regional *hypermethylation*, most of which affects CpG islands and causes, (if these islands are located in gene promotion region), transcriptional silencing of downstream gene. In the context of cancer, methylation affecting tumour suppressor genes is well documented, (see e.g. Jones (2002); Baylin and Ohm (2006)). This is sometimes referred to as *driver methylation*, because of causal involvement in carcinogenesis, (as opposed to *passanger methylation*, which refers to genes whose methylation is a consequence of cancer development, and therefore requires careful analysis of newly detected genes (Ushijima, 2005)).

In the context of gastric cancers, gene inactivation, (e.g. for tumour suppressor gene p16), is more frequently a consequence of aberrant promoter methylation than of defaults of the genetic level (Ushijima and Sasako, 2004), but Ushijima and his team also observed that low levels of aberrant methylation occur in non-cancerous mucosae of cancer patients (Kaneda et al., 2002). This was tested against tissues from healthy individuals, and it was found that methylation levels for these patients were 5.4 to 303-fold higher in *H. pylori*-positive individuals than *H. pylori-negative* individuals (Maekita et al., 2006). This is a very significant finding, since *H. pylori* is known to be a major risk factor for gastric cancers.
It was also highlighted that part of this hypermethylation is temporary and will decrease after eradication of infection. This is not due active demethylation, but to cell turnover. The structure of the gastric crypt, (one stem cell, multiple progenitor cells and many differentiated cells, as shown in Figure 7.3), is the cause for this two-part methylation, where:

- Permanent component is due to methylation of stem cells. Since methylation is conserved during cell division, progenitor and differentiated cells obtained from aberrantly methylated stem cell will exhibit identical abnormal patterns.

- Temporary component is due to methylated in progenitor and differentiated cells which, if stem cell of the crypt is not methylated, will disappear because of cell turnover and creation of new, unmethylated, cells.

Gaining a better insight into these methylation dynamics during infection, (summarised in Figure 7.4a, p.115), and in the long term, (Figure 7.4b), is the objective of the prototype model we have developed, detailed in the following.

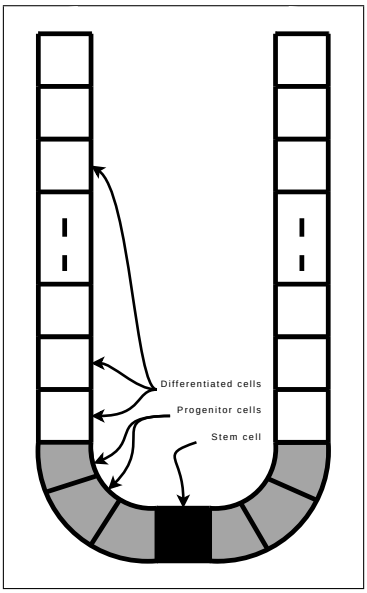
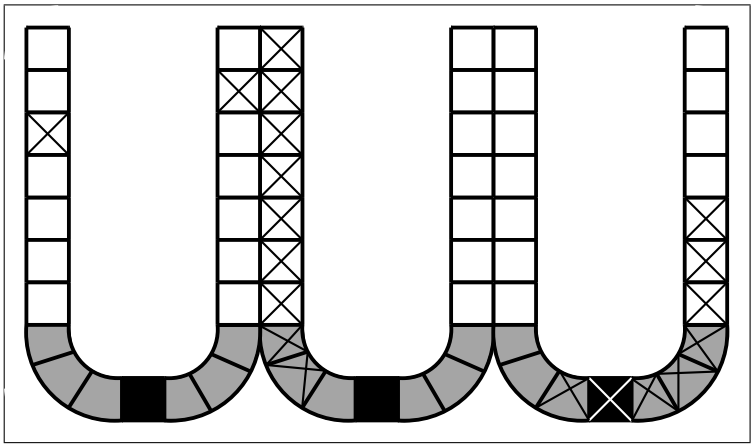


Figure 7.3: Structure of a gastric crypt, with one stem cell, a few progenitor cells and approximately one hundred differentiated cells on each side

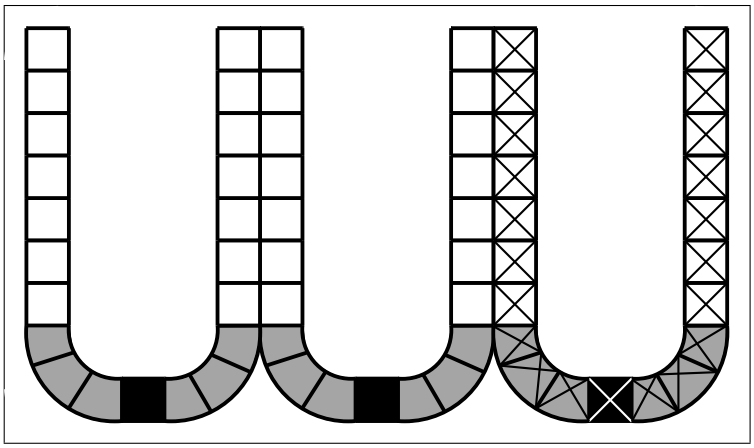
7.4.2 Implementation

To investigate the dynamics of infection-induced aberrant methylation in the crypt, we implement an object-oriented model of the entities involved. The structure of this crypt model is shown in Figure 7.5, (p.117). Key attributes in the model are the infection status of the crypt, its size, the methylation status of each cell, and the overall methylation level in the crypt.

A sample, which mimics *in silico* the samples obtained *in vivo* during NCC experiments,



(a) During H. Pylori infection, cells can be methylated, with different probabilities depending on their type. Methylation status is conserved through cell division, (from stem to progenitor cell, and from progenitor to differentiated cell).



(b) After eradication, only crypts were the stem cell was methylated conserve aberrant methylation, (which is propagated through the whole crypt, during cell divisions).

Figure 7.4: Methylation dynamics in gastric crypts.
Colour represents cell type, (black for stem cells, grey for progenitor cells, white for differentiated cells), and aberrant methylation is shown with a cross inside the cell.

is implemented as an array of crypts. Each crypt is initialised with one stem cell, six progenitor cells, and one hundred differentiated cells on each "wall" on the crypt. No cell is aberrantly methylated.

At each time step, (i.e. every minute of "real time"), each crypt is updated, as follows. First, cells at the top of both "walls" are checked, and removed if they are too old. When there is no infection, their life span is approximately three days. During an infection, it is reduced to almost two and a half days.

Then, we update the bottom of the crypt. A new cell may be created on each side, with a probability p, which is set to approximately 2.3×10^{-3} in normal conditions. During an infection, p is increased, and ranges from 7.0×10^{-3} to 2.6×10^{-2}, as shown in Table 7.1. We then update progenitor cells, which can only produce a limited number of differentiated cells before being replaced by a new progenitor cell.

Finally, we update the methylation status of the progenitor and stem cells. They have methylation probabilities α and β, respectively. The proposed values for α and β are discussed next.

Size increase	Observed frequency (estimation provided by T. Ushijima)	Corresponding range for p
2-3 times	30%	$[7.0 \times 10^{-3}, 1.0 \times 10^{-2}]$
4-5 times	30%	$[1.0 \times 10^{-2}, 1.7 \times 10^{-2}]$
6-7 times	20%	$[1.7 \times 10^{-2}, 2.3 \times 10^{-2}]$
8-10 times	20%	$[2.3 \times 10^{-2}, 2.6 \times 10^{-2}]$

Table 7.1: Crypt size during infection: range of possible sizes, and corresponding values for the probability p to produce a new cell during the crypt update.

7.4.3 Results

The hypothesis from NCC is that, in normal conditions, stem cell methylation is not possible and that, during H. Pylori infection, the probability to methylate these remains significantly lower than that of progenitor cells. As a consequence, the model is built with $\alpha = 2 \times 10^{-7}$ and $\beta = 0$ under normal conditions, and $\alpha = 6 \times 10^{-5}$ and $\beta = 3 \times 10^{-8}$ during infection.

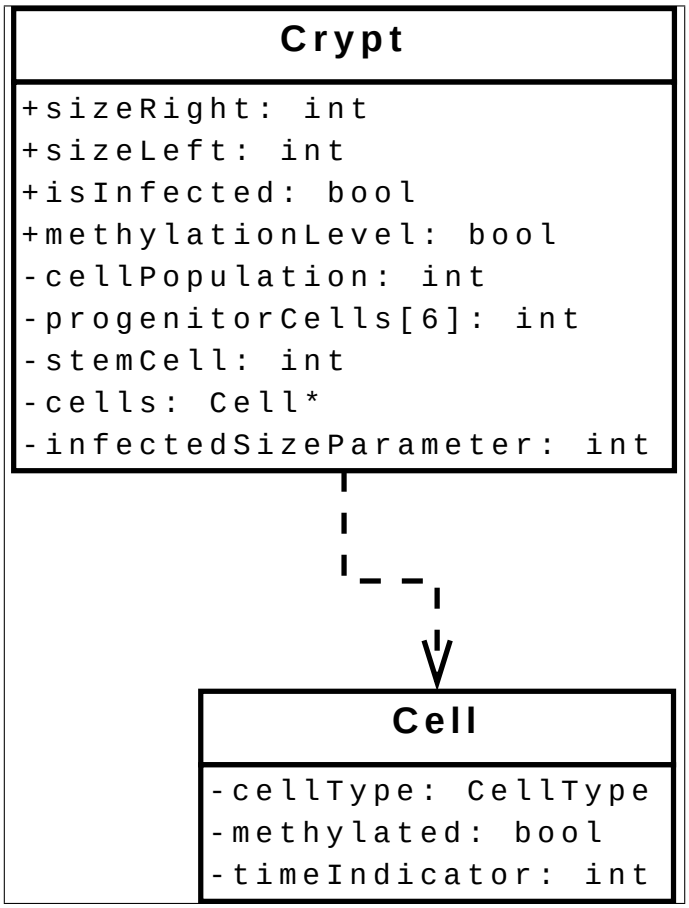


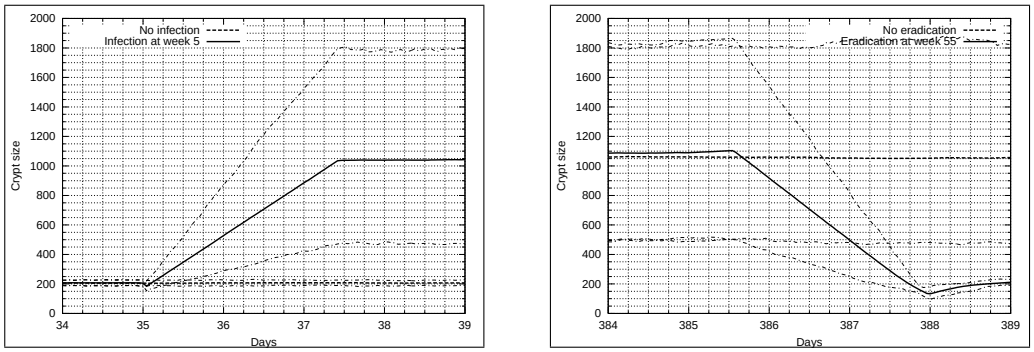
Figure 7.5: Crypt implementation: class diagram

A sample of one hundred crypts is created, and several simulations are performed. Infection, in simulations where it is included, starts at week 5, (i.e. day 35). When H. Pylori eradication is scheduled, this takes place at week 55, (i.e. day 385). This mimics the experimental conditions of the *in vivo* results provided. The results obtained from these simulations are shown in Figure 7.6 for the crypt size, and Figure 7.7 for the methylation levels, (p.119).
It is technically difficult to count cells in the crypt once it has expanded, and quantitative comparison with simulated crypts is, therefore, limited. However, the *in silico* results are, qualitatively, in accordance with the known influence of H. Pylori infection on the crypt. The crypt size initially drops because of the quicker removal of cells near the end of the crypt, but this is rapidly more than compensated by the increased production of cells, and

the crypt expands. The range of sizes successfully reproduces the estimation provided. After eradication, the crypt size starts decreasing after an 18-hour delay, and initially drops below the original size. The normal regime is restored after approximately four days.

The proposed values for α and β confirm the validity of the NCC hypothesis. The probability exists to have infection-induced methylation of progenitor cells 2000-fold greater than that of stem cells, resulting in the two-part methylation. The values for these probabilities in the simulations give ranges for the methylation level in all conditions, (i.e. infection or not, eradication or not), which closely reproduce the *in vivo* results[10] provided.

The model implemented confirms, therefore, both the hypothesis formulated, (since we were able to reproduce it), and the need for *in silico* epigenetic models, (since, in confirming the hypothesis, we provided additional information which was not accessible during the physical experiments, i.e. quantitative values for the methylation susceptibilities).



(a) H. Pylori infection reduces the life span of crypt cells, resulting in initial drop in crypt size, but also induces an increased production, which leads to a significant increase of crypt size.

(b) After eradication, the cells produced in excess are rapidly removed. This results in a crypt size temporarily below normal levels, but this is restored to normal size within days.

Figure 7.6: Simulated crypt size dynamics on a sample of 100 crypts.
Plain and dashed lines correspond to average crypt size, while dot-dash lines correspond to minimal and maximal crypt sizes.

7.4.4 Applications

The model developed is a useful complement to *in vivo* experiments and provides the dynamics of infection-induced aberrant methylation. This is crucial to a better understanding

[10] Confidential results unpublished yet, at the time of writing. A manuscript is in preparation.

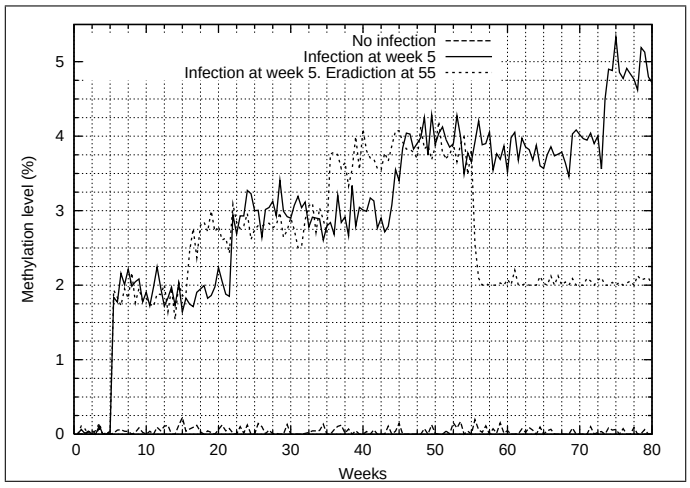


Figure 7.7: Simulated methylation level on a sample of 100 crypts.
Without infection, methylation is almost absent. H. Pylori infection leads to progenitor cell methylation, which results in rapid fluctuations of the methylation level, since these cells have a finite life span and are replaced with unmethylated cells, (unless the stem cell is methylated). The infection also leads, with a smaller probability, to stem cell methylation, which results in permanent and complete methylation of the corresponding crypt, (and therefore a sharp increase of the methylation level of the sample).

of the initiation of gastric cancers.

Methylation levels were observed to be significantly higher in cases of gastric cancer than in healthy volunteers (Maekita et al., 2006), and significantly higher in patients with multiple gastric cancers than in those with a single gastric cancer (Nakajima et al., 2006a). Since patients with multiple cancers are considered to have a higher risk of gastric cancers (Nakajima et al., 2006b), these results demonstrate the potential of methylation levels in non-cancerous gastric mucosae to act as a biomarker for gastric cancer risk.

This is likely to be generalised to other types of cancer: aberrant DNA methylation in non-cancerous tissues was also identified e.g. in the colon (Issa et al., 1994), the liver (Kondo et al., 2000), and the stomach (Waki et al., 2002). *In silico* models can facilitate a quantitative methylation analysis and investigate the influence of potential inducing factors.

Our epigenetic model layer is, therefore, a valuable tool by itself, but is also promising in the context of the main immune model. Epigenetic mechanisms are involved both in normal differentiation of immune cells and in HIV-induced perturbations, (Section 7.3).

Providing data similar to that obtained from NCC for gastric cancers becomes available in the immune context, our model can be applied to these perturbations. The results provided, instead of indicating potential biomarkers as above, would then be used to refine the agent implementation in the lymph node model.

7.5 Chapter summary

In this Chapter, it was demonstrated that it is possible to develop computer-based models of epigenetic changes, and that these models can be a useful complement to physical experiments, both *in vivo* and *in vitro*. Here, the model implemented can be used on its own, or as an additional layer to the immune model detailed in earlier Chapters.

The results from the simulations performed confirm the hypothesis formulated at NCC that H. Pylori infection induces aberrant DNA methylation in progenitor and stem cells of gastric crypts. We were also able to provide estimations for the methylation susceptibility of these cells, and to detail the crypt dynamics during infection, which are difficult to physically monitor.

Progress on epigenetic perturbations in the context of HIV infection will, in the long term, permit the application of this model to HIV-induced aberrant changes, which will enable a refined implementation of the agents in the lymph node model.

Finally, this epigenetic model, being the first to successfully reproduce epigenetic events *in silico*, can be used as a *proof of concept* for further development of more ambitious models, which would incorporate several epigenetic changes and the complex interactions between them.

Chapter 8

Summary and future research

8.1 Summary

The review in Chapter 2 clearly indicates limitations inherent in immune system modelling: as a complex system, a full description by a single model has proven elusive to date, despite some useful insights gained. Commonly, each new model or extension seeks to address known limitations of existing implementations and to provide additional explanation on components of the overall system. In what has been presented here, we have sought to highlight the layers that obtain in building a model for a biological system, where these are characterised by multiple dependencies. Our focus is still improvement of the immune system model with specific entities, but we have also considered exploratory investigations on the layers that sit below this, in particular in our effort to tie in genetic and epigenetic influence. Three layers of complexity are tackled, and for each, limitations are identified and addressed.

The main layer still focuses, at the phenotypical level, on immune response to HIV. To better link known microscopic interactions to observed patterns of disease progression, the use of the agent-based paradigm is advocated and, in that context, several limitations are identified, and addressed.

Four types of agents are modelled which, we believe, fully account for cell-mediated re-

sponse and HIV mechanisms, while this reduces agent diversity to a level which permits simulation of large populations. A short time step, (fifty seconds), is proposed, to guarantee sensible granularity, which ensure that no significant interaction is unaccounted for. Immune memory is implemented at the cell-level, in accordance with principles associated with bottom-up programming. Antigen recognition is also improved, and refined to include adaptability: recognition is not binary.

Further integration at this main layer is provided by linking key activity of lymph nodes in a lymph network. This is crucial, in that it allows modelling of cell mobility both within nodes and between them. This implementation requires, (and permits), parallelisation in order to simulate large network sizes. Details of the implementation are given and where several strategies are presented, tested, and optimised. In particular, a communication strategy, based on a mimic of the lymph network, is identified as the best communication protocol for proposed model.

This has permitted tests on large-scale simulations, as reported in Chapter 6. Known macroscopic signatures of HIV are successfully reproduced. Cell mobility, in particular, leads to accurate representation of viral spread through the lymph network, with a rapid spread within the initial lymph chain, and a slower progression throughout the body. An extension to include localised effects in the gastrointestinal tract is proposed, and details of this new extension are presented and tested.

Additional layers are considered in further chapters and in Appendix A, namely the identification through expression of genes involved in system functionality. Considerations for extracting such information from microarrays are discussed and a new weighting scheme is introduced, and successfully validated using an innovative assessment framework. This scheme is then used in a parallel genetic algorithm which successfully identified co-regulated genes. These results can be applied to all types of microarray datasets and can, in particular, feed into the main layer to refine agent implementation in the lymph node model.

Finally, the sub-genetic model layer is considered, and *in silico* modelling for Epigenetics is presented. Several challenges persist at this level, including overall lack of quantitative

data, but the proposed model of infection-induced aberrant DNA methylation, (target to H. Pylori), is an interesting proof of concept and very promising in terms of future research directions.

8.2 Future research

While the overall multi-layer model addresses significant limitations of existing approaches, it is not exempt from further problems itself. These will be the focus of future research, and are detailed here, for the main layer, (i.e. the lymph network model).

Our first extension would be to consider possible multiple infections of each CD4 cell: a cell can, indeed, be infected by several viral strains. In terms of our model implementation, this should not require important changes. In the current version, CD4 agents can only be infected once. This action is stored using a single integer representing viral strain, but this is easily changed to a list of integers, (as can be found for APC agents and their list of presented antigens). This should not dramatically change model behaviour, but will introduce further variability in disease experience.

In the extension for the lymph network model proposed for inclusion of the gastrointestinal tract, it was explained that not all tissues are lymph nodes. In a first attempt, the generic lymph node implementation was sufficient, but further work may consider refinement of this. For instance, additional tissue may be implemented using smaller matrices and/or different connectivity between these. This will involve further review of the literature in what is a fairly new development, to guarantee optimal implementation.

Cell mobility, as shown in this model, is crucial to both viral spread and immune activation, and has been somewhat intermittently included in previous models. Our current implementation gave an interesting insight into how this affect the overall system, and further refinement is likely be very useful. A few models dealing exclusively with cell recirculation have been proposed (Farooqi and Mohler, 1989; Srikusalanukul et al., 2000; Stekel et al., 1997). Direct integration of these features into our current model is not possible, as

the approaches implemented are very distinct. The results reported, however, can provide a basis for refined local mobility rules.

Another extension would be to explicitly implement thymus operation. In the current version, input of new cells from the thymus is dealt with on a node to node basis, taking into account pressure added by local cell depletion. This is acceptable for the input itself, but neglects possible infection of the thymus itself. Data appears to be relatively sparse for this very local aspect, but some interesting work exists (McCune, 1997). Successful implementation of the gastrointestinal tract is a cause for optimism with regard to the model ability to account for this other localised effect. Further GI extensions detailed above will give further experience as to which architecture is best suited for the thymus implementation, (e.g. in terms of matrix size).

Adding a new type of agent, to account for drug intervention is likely to be very useful. Large-scale simulations may, indeed, give interesting details on when and where treatment should be targeted. This was detailed for the gastrointestinal tract, but is also true with respect to the thymus, and to the overall system. This adds significant complexity, but expertise within the research group can provide insight on techniques to facilitate implementation.

Finally, another extension would be to consider redevelopment of graphical output. As shown in Figure 4.8, (p.58), such a feature was available in early stages of model development, for single-node simulations. Visualisation, however, is non-trivial. In the lymph node model, each matrix element can contain dozens of agents. There is no immediate way to graphically represent an element in which are found two activated CD8 agents, three CD4 agents, one of which is infected, and four APCs! Initial graphical output was not satisfactory, and not adapted to large-scale parallel simulations involving hundreds of lymph nodes. Visualisation can, however, be useful, particularly in terms of dissemination if information to non-Computer Science experts and is certainly worth detailed investigation. Two levels seem indicated. At the lymph node level, initial output may be adapted and would provide an interesting way to demonstrate the impact of cell mobility. At the network level,

a representation of viral spread, e.g. in terms of viral concentration in each lymph node at a given time point, would help to demonstrate growth stages and key targets as well as network vulnerability. This is already under consideration, as we recently outlined (Perrin and Burns, 2008).

8.3 Final remarks

Model layers developed in this Thesis each represent a significant advance to modelling aspects of host response to adverse changes, in that they either address significant limitations of existing models, or provide first attempts in new and promising areas.

Agent-based immune models have now been taken to a new scale, which can account for more than a billion cells and account for crucial phenomena such as cell mobility, as well as localised effects such as early infection in the gastrointestinal tract and, potentially, the thymus.

The contribution to microarray biclustering methods, (Appendix A), is a formalised approach, in that there is a clear and objective framework on which to assess weighting schemes of gene-condition networks, which have been shown to be important in determining gene involvement in specific conditions. Parallel genetic algorithms also provide a powerful way to make the most of these schemes, including the one proposed here.

Epigenetic modelling is still, of course, at a very early stage, but the initial model is a significant first step towards a better understanding of involved phenomena. Attempts to integrate several layers in human system-based modelling also represent a significant development, and this Thesis, we contend, provides an important contribution to the Computational Biology field.

Appendices

Appendix A

Beyond phenotype: understanding gene expression

A.1 Objectives

The mechanisms leading to the development of the phenotype form a biological system at least as complex as the immune response itself studied in the main body of this Thesis, and the objective of this Chapter is not to provide a definitive description and model for this system. Rather, it is to propose tools for a more efficient analysis, (and indicate how these can be built in the main model). The results obtained from this enhanced analysis can indeed, in the future, lead to a better understanding of gene function in the emergence of immune responses. This would form an *additional layer* to the current main model, leading to refined agent implementation. Here, a *layer* does not mean using these results directly during the simulation, but rather refers to a layer of *knowledge* used during model development. Gene function analysis in the context of the immune system is a novel area, (see e.g. (Sarson et al., 2007)), but is very promising.

The link between these two layers is shown in Figure A.1, (p.129), but a better understanding of gene function can of course be applied to several areas, and is not limited to HIV modelling. Common techniques used to study gene expression include microarrays and

related analysis techniques. Here, we chose biclustering to extract valuable information from microarray datasets. This new layer therefore uses low-level information, (levels of expression of thousands of genes), and produces refined data at the same modelling level, (identification of genes with similar expression pattern). This classification of genes is useful for all fields of biology, (immunology or otherwise). In the particular context of our immune system model, such analysis, performed on microarrays specifically dealing with immune cell lineages, can be used to refine the lymph node model, through a more detailed implementation at the agent level.

Analysis methods for gene expression microarrays, even though widely used, often lack formal validation and evaluation, either as a whole, (as outlined in Turner et al. (2005)), or even of their main components. In particular, these techniques rely heavily on weighting schemes which, given their importance, are surprisingly rarely analysed. This is also true for algorithms underpinning the analysis, yet most are said to perform "reasonably". This presents difficulties in terms of meaningful evaluation and attempted improvements. Some basic questions include:

- Is this technique efficient thanks to a well-designed weighting scheme? If this is the case, then this scheme is portable, and efforts should focus on improving the analysis algorithm.
- Is the analysis technique relying on a robust and efficient analysis algorithm? In that case, the focus should be on improving the weighting scheme.

In some cases, both components of the analysis may represent considerable advances for the field. It is also true that not all techniques work equally well under all circumstances. However, a poor technique overall could still have, as one of its components, an interesting weighting scheme or analysis algorithm.

A crucial objective of this Chapter is, therefore, to introduce evaluation of both analysis components. A new weighting scheme and parallel genetic algorithm for biclustering are presented and evaluated.

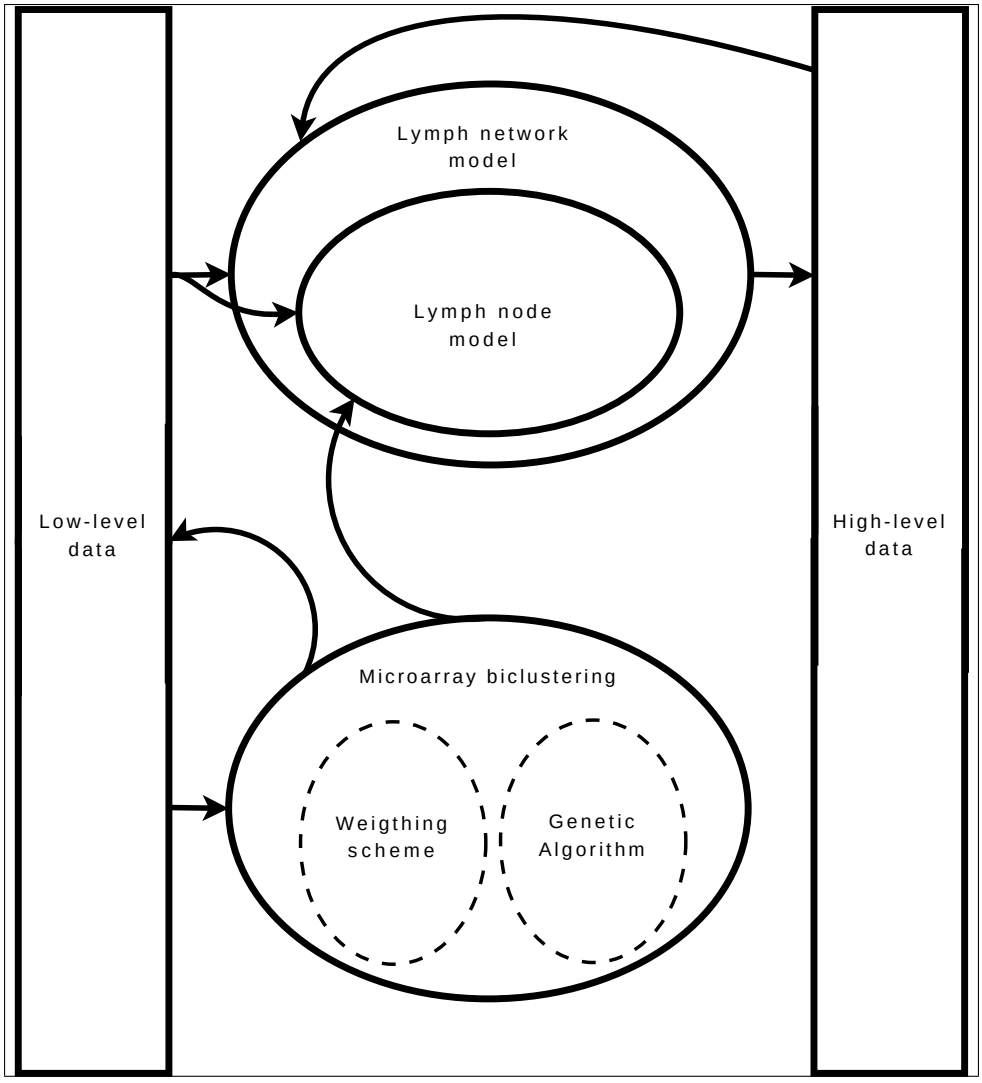


Figure A.1: Biclustering as an additional layer of the overall model.

Low-level biomedical information, (i.e. level of expression of thousands of genes) is used as an input to this layer. The results from the analysis, (identification of genes with similar expression pattern), can be used directly by biologists, or can be used in the lymph node model to refine the agent implementation and, in turn, enhance the realism of the immune system model. A biclustering technique requires two components: a weighting scheme and an analysis algorithm. Here, a genetic algorithm is chosen for the analysis.

A.2 Gene expression microarrays

A.2.1 Measuring expression levels of genes

Microarray technologies are used for large-scale transcriptional profiling, through measurement of expression levels of thousands of genes at the same time. The motivation here is that by understanding gene expression, further insight will be gained into cell function and cell pathology (Valafar, 2002).

Expression-intensity values, (based on fluorescent techniques), are recorded for multiple microarray experiments carried out under several conditions, (e.g. environmental, biological phases, different biological tissues). The data obtained is often presented as a real-valued matrix: a row contains the expression pattern of one gene over all the conditions, while a column represents the pattern of expression of all genes for one condition. Each matrix element X_{ij} is, therefore, the measured expression of a gene i under condition j.

Analysis is needed, to extract meaningful information, from the large datasets, about the system being studied. Given the amount of data produced, this is not trivial.

A.2.2 Microarray biclustering

Several techniques have been developed over the years to analyse gene expression microarrays, (see e.g. Stolovitzky (2003) for a general review), and several of these are variations of the concept of clustering. The objective here is to group genes, based on their expression under multiple conditions (or over different time-points) or, conversely, to group conditions according to expression of several genes (Raychaudhuri et al., 2001; Slonim, 2002).

Biclustering was introduced, by Cheng and Church (2000), as the simultaneous clustering of both genes and conditions. The authors identify three advantages over traditional clustering:

- Biclustering is better at selecting genes and conditions with more coherent measurement and dropping those representing noise.
- Grouping of genes through biclustering is based on similarity in the context of the

subset of conditions. Biclustering, therefore, discovers both grouping and context, a result more advanced than that obtained from successive clustering of rows and then columns separately.

- Biclustering allows genes and conditions to be part of multiple biclusters, i.e. be identified by more than one functional category. This is reflective of actual functionality of genes.

These authors also considered the NP-hardness of the problem, later shown to be NP-complete (Peeters, 2003).

Several categories of biclustering algorithms coexist, and vary as to type of biclusters achieved. While some look for biclusters with constant values on rows and/or columns, (see e.g. Busygin et al. (2002)), it was highlighted, in a recent survery (Madeira and Oliveira, 2004), that more advanced, improved algorithms locate biclusters with coherent values (Wang et al., 2002) or coherent evolution (Liu and Wang, 2003). This latter category is very interesting, since biclusters in this case are formed irrespective of the exact expression values, but rather by looking at evidence that a group of genes show similar expression patterns, (up-regulated, or down-regulated), over a number of conditions. A widely-used algorithm from this category is SAMBA[1] (Tanay et al., 2002).

However, it is difficult to comprehensively evaluate existing tools when parts of the analysis technique are not validated individually. The remainder of this Chapter will, therefore, focus on development and analysis of the elements of a biclustering algorithm built on a new weighting scheme and parallel genetic algorithm.

A.2.3 Problem formulation

The first of the proposed technique is to consider data from the gene expression matrix as bipartite graph. Two sets of nodes represent genes and experimental conditions, respectively. Edges are limited to connecting these two sets, hence the bipartite structure. It is a complete

[1] SAMBA: Statistical-Algorithmic Method for Bicluster Analysis.

bipartite graph (or *biclique*), since there is an edge linking any pair of gene-condition. An example of a complete bipartite graph is displayed in Figure A.2.

Weights are then assigned to edges, based on expression data stored in the matrix, so that

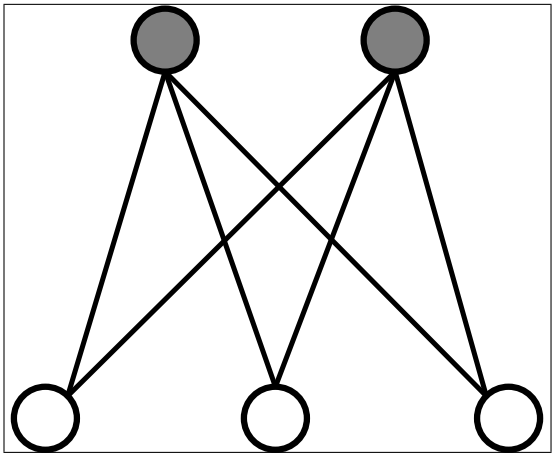


Figure A.2: A complete bipartite graph with partitions of size 2 and 3

gene-condition pairs identified as "interesting" are given negative weights. In this context, a bicluster is a subgraph which conserves the biclique structure. In other words, the proposed algorithm will look for bicliques with minimum total weight.

Since the search domain is a complete bipartite graph, the biclique structure can be described as follows (and is summarised in Equation A.1:

- An edge can not be included in the biclique if the associated gene is not included.
- An edge can not be included in the biclique if the associated condition is not included.
- If a gene and a condition are included in the biclique, then the edge linking them together is also included.

The biclustering problem is, therefore, mathematically defined by Equation A.2. This problem being NP-hard, the mathematical formulation can not be used as is for large microar-

rays, and a meta-heuristic method is implemented.

$$\begin{cases} x_{ij} \leq y_i & \text{(edge not in the bicluster if gene not included)} \\ x_{ij} \leq y_j & \text{(edge not in the bicluster if condition not included)} \\ x_{ij} \geq y_i + y_j - 1 & \text{(edge in the bicluster if both are included)} \\ x_{ij} \in \{0,1\} & (x_{ij} \text{ is a boolean variable}) \\ y_i, y_j \in \{0,1\} & (y_i \text{ and } y_j \text{ are boolean variables}) \end{cases} \tag{A.1}$$

$$Minimise \sum_{i,j} c_{ij} x_{ij} \tag{A.2}$$
$$\begin{cases} \text{conditions (A.1)} \\ x_{ij}, y_i, y_j \in \{0,1\} \\ c_{ij} \in \mathbb{R} \end{cases}$$

A.3 Weighting schemes for microarrays

A.3.1 Importance of weighting schemes in analysis techniques

As highlighted in Section A.1, weighting schemes are a crucial element of microarray analysis techniques. All biclusters found by a particular technique are of course dependent on the weights used to evaluate them. Different weights, all else being equal, provide different results and interpretation of gene expression patterns.

A review of existing methods leads to the observation that there are almost as many weighting schemes as there are analysis techniques, since most new techniques incorporate a new weighting scheme, or a variant of an existing one.

Yet, these schemes are rarely explicitly validated, and results are often restricted to the overall method. It is difficult to estimate whether this is a cause, or a consequence, of the abundance of schemes, and it may appear that a new scheme is the last thing that is needed.

Nevertheless, in the remainder of this Section, a new weighting scheme is introduced. This scheme is technique-independent, which seems the best way to break the current "one technique - one scheme" rule for microarray analysis. This scheme is evaluated through a set of criteria and, later in the Chapter, we use it ourselves as a key part of the biclustering algorithm.

A.3.2 Development of a new weighting scheme

The underpinning motivation of the distribution-based scheme proposed here is that it is difficult, if not impossible, to give an absolute characterisation of an interesting, and biologically significant, gene-condition couple. An absolute expression level, on its own, means very little. On the other hand, the study of the expression level of a given couple relatively to that of other related couples could give an interesting insight. A biologically meaningful couple is one where we can highlight a significant effect of the condition.

It implies an expression level differing notably from the average expression pattern observed for this gene over all the conditions. It also implies some deviation from the effect this condition has on the whole set of genes: if a certain condition leads to over-expression of virtually all the genes, it might not be very interesting to consider its effect on one of these many genes.

In this approach, the microarray is considered as a matrix containing only positive values: the expression levels. Some microarray datasets are only made available after they are transformed into log space, (a result of the normalisation process), thus leading to some negative values for low expression levels. For such cases, the first step is to transform the data back from log space, to deal exclusively with positive values.

Actual expression levels are then replaced by *expression ratios*, obtained by dividing the expression level of one gene under one condition by the average expression level of this gene over all the conditions. For a given gene, we therefore obtain a series of positive values, of average unity, with values below unity when the gene is under-expressed under a certain condition, and greater than unity for a gene over-expressed under the condition.

The next step is used to account for the second aspect detailed above. For a given condition, we want to differentiate between the genes that have a specific behavior and those that react to the condition as most other genes do. This is achieved using a geometric series to create categories with a small width for values close to unity, and increasing as ratios are further from this value. Such series are used because of the skewness of the data: many genes show very little response to a given condition and therefore having an expression ratio close to unity. Two series are used: one for ratios greater than 1, and the other for ratios smaller than unity. The common ratio R for each geometric series is calculated using Equation A.3, where *categories* is the number of categories needed, and *Min* and *Max* are, respectively, the smallest and the greatest value of the partial set considered for the series. For the $[1; \infty)$ part of the dataset, $Min = 1$, and the category boundaries are, therefore, 1, R, R^2, etc., with $R \geq 1$. For the $[0; 1]$ part of the dataset, $Max = 1$, and the category boundaries are, therefore, 1, R, R^2, etc., with $R \leq 1$. For this part of the dataset, there can be a problem if $Min = 0$, since R cannot be calculated on such cases. To prevent this from happening, we only consider values different from 0 to compute Min.

$$R = \left(\frac{Max}{Min}\right)^{1/categories} \tag{A.3}$$

Once the categories have been created and populated, weights are given to the gene-condition couples, depending on the size of the category they belong to. As most optimization techniques are traditionally used to minimize a given objective function, we want to have negative weights for "interesting" couples, and positive ones for the others. The easiest way to obtain size-dependent weights following this rule is to subtract the average population of a category from the population of the category the couple belongs to: "small" categories get negative weights, while the bigger ones get positive values. To avoid unnecessarily large weights as the number of genes in a dataset increases, weights are then normalised using this number of genes.

A.3.3 Validation and analysis

A.3.3.1 Introducing an assessment procedure

As explained before, it is crucial that weighting schemes are validated on their own before being used in a biclustering algorithm. An assessment procedure is needed. Discussions with Gráinne Kerr[2], (see e.g. Kerr et al. (2008) for previous experience on microarrays analysis), resulted in the five properties proposed below:

- Discrimination. Is the discrimination between "good" and "bad" gene-condition couples significant enough? Is the weighting scheme introducing *false positives* or *false negatives* in terms of interesting couples?
- Robustness. Is the scheme's response to noise and missing values reasonable?
- Configurability and parameter influence. How flexible is the scheme, and how does this flexibility influence discrimination and robustness?
- Reusability. Is the proposed scheme effectively technique-independent?
- Biological meaning. If the weights can be interpreted biologically, discrimination and reusability are increased, and validation is made easier.

A.3.3.2 Analysis of the proposed weighting scheme

The proposed scheme has some flexibility, through choice of the number of categories, k. The influence of that parameter will be assessed through examination of the robustness and discrimination achieved. The scheme is also technique-independent, since it was designed without any particular subsequent analysis algorithm in mind. The only restriction here is that microarrays are considered as bipartite graphs, but this representation is common in current biclustering techniques, and does not limit the reusability of the scheme.
Biological meaning of the weights is a direct consequence of the weighting process: the lower the weight, the more significantly the expression level of the gene deviates from that

[2]PhD Student. School of Computing, DCU, Ireland.

observed for the majority of genes.

Several techniques exist to reduce noise or deal with missing values from microarray experiments, (see e.g. Adjeroh et al. (2006); Verboven et al. (2007)). The objective here is not to review techniques used to deal with them, but to evaluate how the scheme reacts to residual noise and missing values. With this in mind, only low perturbations should be expected, and the scheme is tested for perturbation levels up to 10%.

The influence of noise is summarised in Table A.1, for tests on the Gefitinib Treated Kasumi Cell Line Dataset[3]. Similar results are obtained for two other datasets: the Yeast Cell Cycle[4] and the Lymphoma dataset[5]. Influence on the weights is reasonable for low noise perturbation. "Stable" weights are those for which the variation is smaller than the noise added to the dataset. It is an interesting indicator, because of the nature of the scheme: when a gene-condition couple falls into a new category due to added noise, this changes the weights for all couples in the new category as well as all those in the previous one. Using this value may, therefore, give more insight into the scheme robustness, than using only the average absolute variation of the weights. Here, it confirms the behaviour is satisfactory for expected levels of noise perturbation. The evolution of that proportion of stable weights when the number of categories changes is displayed in Figure A.3a (p.140). Clearly, as the number of categories increases, the width of each category decreases, and gene-condition couples are more likely to change categories, leading to a smaller proportion of stable weights. However, the average absolute variations are almost unchanged, (Figure A.3b, p.140), so the effect of a larger number of categories on the weights assigned is relatively small.

The influence of missing values is summarised in Table A.2, for the same dataset. Again, the patterns obtained with the other datasets are similar. With respect to missing values, the scheme is far less robust than in response to noise. Notably, the sign of the weights is not

[3]avalaible from the MIT Broad Institute website, `http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi`.

[4]available from R.W. Davis' website at Stanford, `http://genomics.stanford.edu/yeast_cell_cycle/cellcycle.html`.

[5]available from the Lymphoma/Leukemia Molecular Profiling Project Gateway, `http://llmpp.nih.gov/lymphoma/`

Noise level	0%	1.5%	2.5%	5%	10%
Average absolute variation	0%	4.26%	6.47%	11.8%	21.6%
Proportion of "stable" weights	100%	84.2%	86.4%	86.5%	82.0%

Table A.1: Influence of noise (for 20 categories)

lost: a positive weight does not become negative, (except for cases when more than half the values for a given gene are missing, but in those cases, it would almost certainly be excluded from the dataset before the scheme is applied). Biological significance is, therefore, conserved. What is partially lost is the degree of over-expression, (or under-expression), rather than the knowledge that this change of expression occurs. The evolution of the influence of missing values depending on the number of categories is displayed in Figure A.3c (p.140) for the average absolute variation, and in Figure A.3b for the proportion of stable weights.

As noted previously, a range of techniques dealing with missing values are not considered here, but an extended evaluation of the scheme must account for these. Indeed, the results displayed correspond to untreated data, expected to create large perturbations in the weights: for these missing values, an expression ratio of unity is artificially created, (a very simple, conservative, and certainly very poor, correction technique). Future work might reasonably include tests of the effect of the various correction techniques on the scheme robustness.

Proportion of missing values	0%	1.5%	2.5%	5%	10%
Average absolute variation	0%	23.7%	38.5%	74.2%	141%
Proportion of "stable" weights	100%	35.28%	35.7%	32.6%	30.1%

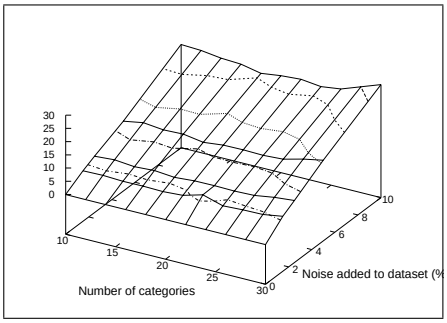
Table A.2: Influence of missing values (for 20 categories)

The last series of tests consider the discrimination between gene-condition couples and examines performance in terms of biological meaning. By construction, there is no "damaging" false-positive or false-negative, at least in theory. In practice, these may occur for weights very close to zero, (either positive or negative, depending on the number of cate-

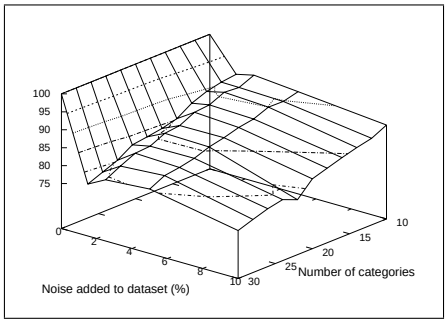
gories), but would not have a significant impact on biclustering. This, because these weights are so close to zero, and correspond to "neutral" gene-condition couples. Not all gene-condition couples with negative weight have to be included in any bicluster, of course, but a significant change in expression patterns can not lead to a positive weight, while negative weights are only obtained where there *is* a significant change. Since *absolute*, (positive vs. negative), discrimination is guaranteed, the focus here is to assess *relative* discrimination, i.e. the distribution of weight values. Results on the influence, of the number of categories, on this discrimination are displayed in Table A.3. A first observation is that discrimination is good with, on average, just under 25,000 negative weights (out of 222,830). This is consistent with the biological context: most genes are not *specifically* reacting to any given condition. This value varies between 23,170 and 28,392, with a standard deviation of 1,682. Discrimination is, therefore, very satisfactory for any number of categories in the range tested. Weights obtained with fewer categories appear more refined, (except for 10 categories, which seems to indicate that going lower would be ill-advised). Given that this range also corresponds to improved robustness, using $k = 12$ to 16 categories is recommended. This recommendation also applies to the other datasets, for which the results obtained are similar.

Weights	$\leq$-4	[-4;-3]	[-3;-2]	[-2;-1]	[-1;0]	[0;1]	[1;2]	[2;3]	≥ 3
10 categ.	5035	5630	4838	6230	6659	7327	4237	1772	181102
12 categ.	739	6241	5788	8217	5075	9371	6070	6010	175319
14 categ.	0	3210	7330	7603	7474	9197	7979	7957	172080
16 categ.	0	615	7823	8076	10137	6684	9168	11244	169083
18 categ.	0	0	6489	9283	7866	10708	7696	14108	166680
20 categ.	0	0	4318	9392	10350	10597	8686	11288	168199
22 categ.	0	0	2239	9834	11621	10975	11912	8494	167755
24 categ.	0	0	666	10747	12161	13237	8018	12875	165126
26 categ.	0	0	0	10806	13369	11863	10451	14696	161645
28 categ.	0	0	0	9369	13801	14670	11029	15410	158551
30 categ.	0	0	0	8266	15102	14362	13409	14972	156719

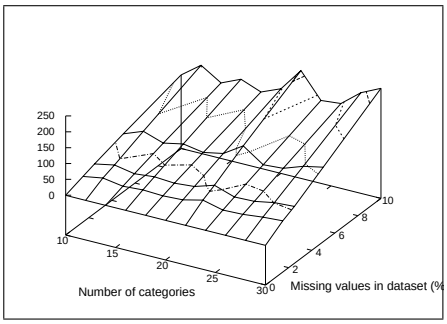
Table A.3: Influence of the number of categories on discrimination: distribution of weight values



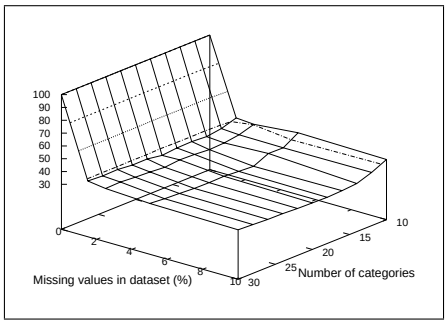
(a) Influence of noise - Perturbation levels (%)



(b) Influence of noise - Proportion of stable weights (%)



(c) Influence of missing values - Perturbation levels (%)



(d) Influence of missing values - Proportion of stable weights (%)

Figure A.3: Evaluation of the robustness of the weighting scheme: influence of noise and missing values

A.4 Biclustering through parallel genetic algorithms

A.4.1 Genetic algorithms and their application to microarrays

A genetic algorithm is an optimisation technique that was loosely inspired by mutations in nature and how these lead to biological evolution through survival of the fittest elements only (Holland, 1975).

The first step is to organise coordinates of points in the problem space as a sequence, inspired by gene sequences. A population of sequences is created and a search for optimal solutions with respect to a fitness function is accomplished by mutating the sequences, hence allowing transformation to new coordinates in the problem space. Each new sequence is evaluated, to determine whether it represents a new optimum. Other techniques have been added to mutations to create new sequences, as will be detailed when presenting the imple-

mented algorithm.

Genetic algorithms have been extensively used in the context of biological applications, such as DNA fragment assembly (Cedeno and Vemuri, 1993; Fickett and Cinkosky, 1993; Parsons et al., 1995), multiple molecular sequence alignment (Zhang and Wong, 1997) and phylogenetic analysis of proteins (Hill et al., 2005).

For microarray biclustering in particular, there have also been some attempts at developing genetic algorithms (Chakraborty and Maka, 2005; Mitra and Banka, 2006), but assessment of these implementations is limited, (Section A.1).

Moreover, a well-known limitation of genetic algorithms is that a large population of sequences is required (Goldberg and Deb, 1991; Jackson and Norgard, 2008). A solution is to consider parallel implementations, and to the best of our knowledge none has been developed for biclustering. The objective of this Section is, therefore, to *introduce and validate a parallel genetic algorithm* for biclustering of gene expression data from microarrays.

A.4.2 Parallel genetic algorithms

The parallel nature of genetic algorithms has been considered from the start and several early implementations have been proposed (Grefenstette, 1981). Since then, different approaches have emerged. As proposed by Cantu-Paz (1995), we can categorize these as follows:

- Global parallelisation. Evaluation of solutions and genetic evolution of the population are explicitly parallelised, and each solution has a chance to combine with any other.

- Coarse grained parallelisation. Population of solutions is divided into subpopulations, and these are isolated from each. To deal with these, this implementation introduces a migration operator. Two types of implementation coexist in this category. In the *island model* individuals can migrate to any other subpopulations, while in the *stepping stone model*, migration is limited to neighbouring ones.

- Fine grained parallelisation. Subpopulations are very small, ideally only one solution

is run on each processor. This, of course, requires a massively parallel computing architecture.

- Hybrid parallelisation. The three previous strategies can be combined.

The most popular strategy is to use coarse grained parallelisation, (see e.g. Levine (1994); Pereira and Lapa (2003)), and several implementation challenges have been reported (Cantu-Paz, 1995; Katayama et al., 2003). These include:

- Topology. Connectivity of subpopulations affects convergence. Balance is required between isolation, which allows development of new solutions, and efficient mixing, which leads to propagation of good solutions.

- Migration rate and frequency. Again, balance is required between sharing too many solutions, or too often, and not having a sufficient mixing, which would lead to independent runs of genetic algorithms on small populations, producing poor results.

- Size of subpopulations. Larger samples mean better results, but also imply longer computation time.

- Effectiveness of genetic operators.

With these in mind, the following algorithm has been implemented and tested.

A.4.3 Algorithm development

A.4.3.1 Parallel structure

The proposed architecture is coarse-grained parallelisation based on the stepping stone model. For migrations, each subpopulation is sorted according to the total weight of the encoded bicluster, and a bidirectional ring is used: solutions travelling clockwise are selected from the "rich area", (which contains the best solutions of the subpopulations, i.e. solutions with lowest total weights), while solutions travelling anti-clockwise are selected from the "poor area", (which contains the solutions with the highest total weights). "Rich"

and "poor" areas of a subpopulation are defined using threshold values for the total weight of the encoded bicluster. This topology is detailed in Figures A.4 and A.5.

This parallel structure has six specific parameters:

- Number of subpopulations, s_1, and size of each, s_2.
- "Rich" area threshold, r. If a bicluster weighs less than r, (we are minimising), it is considered a "rich" solution.
- "Poor" area threshold, p. If a bicluster total weight is greater than p, it is considered a "poor" solution.
- Number of local iterations between two migration steps, n.
- Number of migrants sent in each direction at each migration step, m.

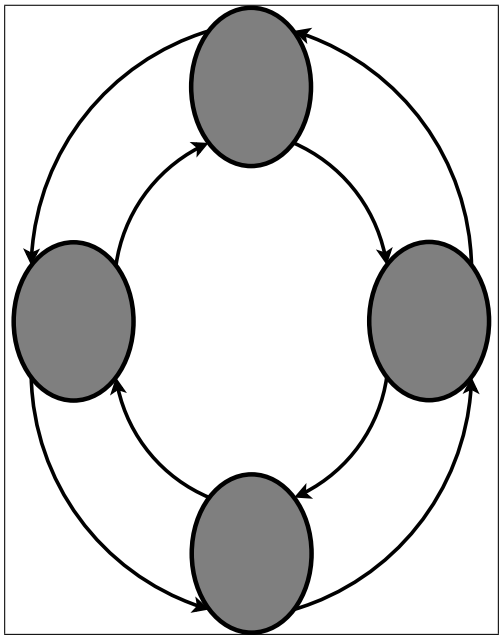


Figure A.4: Coarse-grained, stepping stone structure.
Here, four subpopulations are connected through a bidirectional ring. Selected solutions are sent clockwise if they correspond to biclusters with a low total weight, and anti-clockwise if they correspond to biclusters with a high total weight

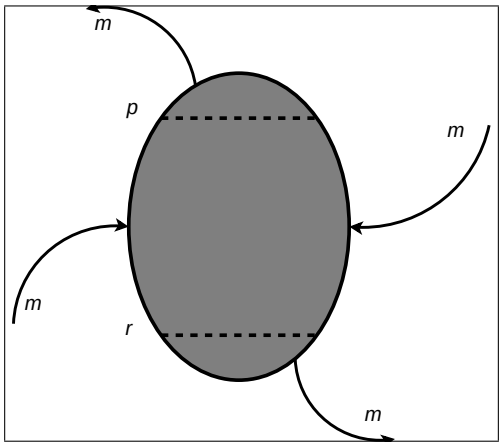


Figure A.5: Parallel topology.
Before a migration step, each subpopulation is sorted. Then, m solutions are selected from the most promising ones, (i.e. those ranked below threshold r), and are sent clockwise. Similarly, m solutions are selected from the least promising ones, (i.e. those ranked above threshold p), and are sent anti-clockwise. Finally, new solutions are received: m good solutions arrive clockwise, and m poor solutions arrive anti-clockwise.

A.4.3.2 Local genetic algorithm

Based on this parallel structure, the local genetic algorithm is developed, in collaboration with Christophe Duhamel[6], (who has previous experience on genetic algorithms, see e.g. Potvin et al. (1996)).

Encoding the solution

The first step is to consider solution encoding as "genes". Even though more advanced encodings are sometimes proposed (Chen et al., 2006), the traditional approach, chosen here, is to encode solutions as genes using binary variables. Given the objective function defined in Equation A.2, a naive solution would be to used edge presence in the solution bicluster as a Boolean variable. The obvious limitation in this case is the length of the resulting array: with 20,000 genes and 10 conditions in the dataset, (which is not unusual, as the validation of the weighting scheme highlighted), 200,000 boolean variables are needed for each solution! A second approach would be to have an array encoding presence of genes and conditions in the solution bicluster. An important observation in this case is the relative

[6]Lecturer. Laboratoire d'Informatique et de Modélisation des Systèmes, ISIMA, France.

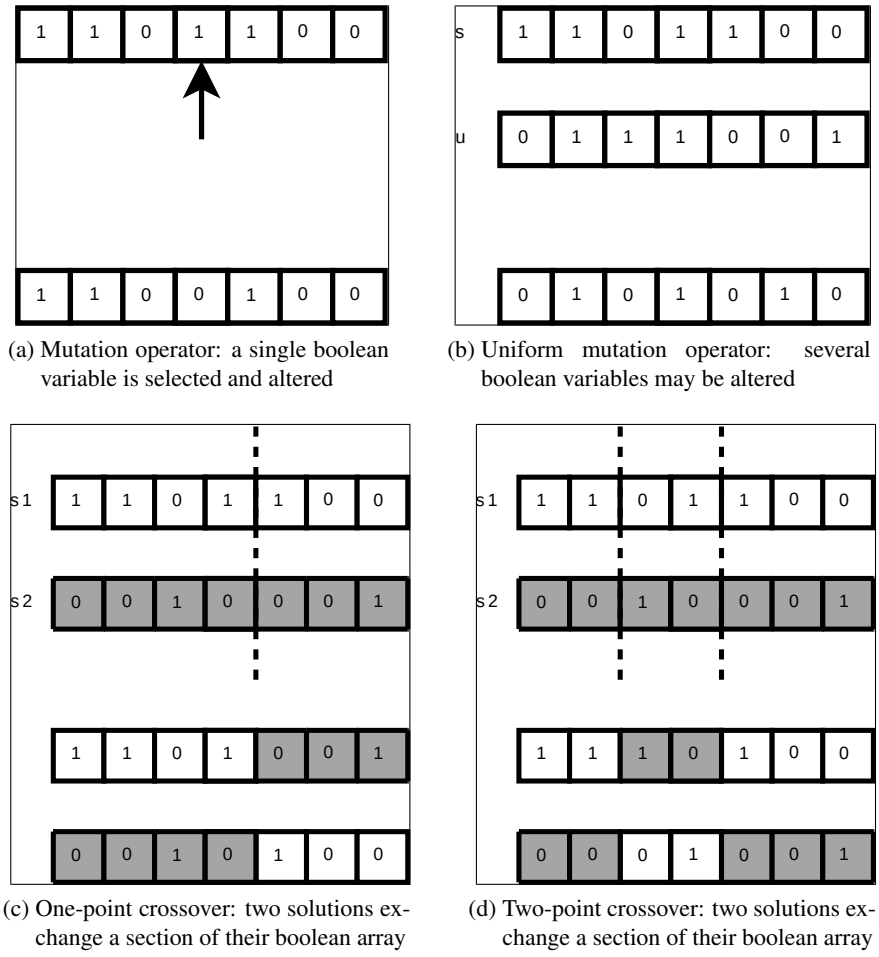
length for each part of the resulting array: there are often thousands of gene in a microarray dataset, while finding a hundred or more conditions is very rare. The consequence is that evolution operators will, statistically, mostly concern parts of the array corresponding to genes, rather than conditions. Interestingly, *explicitly encoding genes is not even necessary*. Indeed, once a subset of conditions is chosen, interesting biclusters only involve genes for which the total weight over the selected conditions is negative. It is, therefore, possible to perform biclustering while explicitly encoding only *a small part* of the bicluster.

Evolution operators

Evolution operators are used to increase the population of solutions, by introducing new solutions obtained through small variations of existing ones. Since earliest implementations, a characteristic operator is *mutation*. A solution is randomly chosen, and one of its boolean variables is altered. This is summarised in Figure A.6a. A more drastic operator, based on mutation, is also implemented in the proposed algorithm: the *uniform mutation*. A solution s is randomly chosen, and a boolean array u, of same length, is also created. A new solution is then obtained by conserving the value of a boolean variable $s[i]$ where $u[i]$ is equal to 1, and altering it otherwise, as shown in Figure A.6b. This operator induces more diversity than traditional mutation, but can also degrade a good solution. It is, therefore, recommended not to use it as frequently as the first one.

Another consequence of the *bio-inspired* nature of the algorithm is the use of *crossover operators*, which are loosely based on the biological phenomenon occurring during *meiosis*. Here, two existing solutions are chosen. These can be selected a uniform probability or it can imposed, for instance, that one of them must belong the best 10% solutions. This second approach is taken here, to ensure further "investigation" of promising solutions. Once the two solutions are chosen, a cutting point is selected in the solution array, using a uniform probability, and the solutions exchange the variables located after that point, as shown in Figure A.6c. A variant exists where two cutting points are chosen, and solutions exchange variables located between those two points, as detailed in Figure A.6d.

Selection



(a) Mutation operator: a single boolean variable is selected and altered

(b) Uniform mutation operator: several boolean variables may be altered

(c) One-point crossover: two solutions exchange a section of their boolean array

(d) Two-point crossover: two solutions exchange a section of their boolean array

Figure A.6: Evolution operators for the "expansion" phase of the algorithm

A typical iteration of a genetic algorithm includes the creation of new solutions, (i.e. the "expansion phase"), followed by the evaluation of population and selection of solutions that will be conserved for the next iteration, the remainder being eliminated, (i.e. the "selection phase"). Traditionally, between these two phases, it is necessary to consider "repair" functions, which will restore validity of the new solutions created through evolution operations. Here, because of the chosen encoding, any array is a valid solution, and such functions are not required.

The final step is, therefore, to consider the "selection phase" of the algorithm. Here, several approaches can be taken. In a first one, deterministic selection is used: if the initial population was n, then the n best solutions in the expanded population are kept, and the

other eliminated. The main advantage here is that, once the population is sorted, this type of selection has a very low computing cost. Diversity, however, may be damaged. The alternative approach is to consider "tournaments" between solutions, with the winner kept and the loser eliminated. In this case, selection is obtained as follows: while solutions still need to be removed, two elements within the current population are selected, and the one with the best fitness value is chosen to be conserved with a probability p. This probability is used to adjust the selection pressure: $p = 1$ is equivalent to the deterministic selection described above, while $p = 0.5$ would correspond to a uniform selection which would not take fitness into account.

Here, a hybrid approach is taken: the population is sorted, the $n/2$ best solutions are conserved, while the others are involved in 'tournaments" until we obtain a population of size $9n/10$. Population size is then restored to n by introducing newly created solutions. Each of these solutions is created as follows: (i) for each condition j, we count k_j, the number of times it appears in the N current solutions[7]; (ii) we generate random numbers r_j, uniformly in $[0, N]$; (iii) condition j is included in the new solution if and only if $r_j > k_j$. This improves the diversity in the overall population.

A.4.4 Validation of the genetic algorithm

A.4.4.1 Objectives and framework

The proposed genetic algorithm is implemented, and tested on a cluster architecture. The objective of these tests is to determine whether, given a specific set of weights, the algorithm can isolate useful biclusters. The results of these tests are detailed here.

To do this, it is, of course, necessary to extract these biclusters from the set. Mathematically, it is possible, for a microarray dataset with m conditions, to find the best bicluster using k conditions. Doing this, for all possible values of $k \in [0, m]$, will extract these biclusters. The main limitation here is, of course, the number of potential biclusters. In a microarray with m conditions, there are 2^m possible subsets of conditions, and the computation time of

[7] with $N \in [9n/10, n]$.

this exact method is, therefore, proportional to this value. With 10 conditions, this method takes approximately half-a-second. This gives a computation time of the order of 2^{m-11} seconds. This corresponds to just over a minute with 17 conditions, four and a half hours with 25 conditions, and already several thousand years with 50 conditions. This method is obviously not practical, but for small microarrays, it offers the means to assess the genetic algorithm.

A.4.4.2 Performance on small microarray datasets

Validation of the genetic algorithm is, therefore, performed on the two microarray datasets used for assessment of the weighting scheme. Table A.4 shows results obtained from the enumeration method and from the genetic algorithm, for those two datasets. On the KCL dataset, the genetic algorithm performs very well, and even a single run of the local implementation with a population of size equal to the number of conditions, (here, $n = k = 10$), finds the best solution. On the YCC dataset, however, the same local implementation is more limited and some solutions it provides are quite far from the optimum. The parallel implementation, (with local parameters unchanged and sixteen islands), still provides optimal solutions. To further demonstrate the interest of this parallel implementation, severals runs of each implementation are performed. Results are shown in Table A.5. The local implementation finds each optimal solution in at least 10% of all runs, but just under half of them are identified every time. The parallel implementation, with a similar computation time, finds the best solutions every time.

A.4.4.3 Performance on large microarray datasets

For larger datasets, the exact method for extraction of the best biclusters can not be used, and the results from the genetic algorithm can not be compared to known optimal solutions. The alternative is to use a heuristic to identify "good" biclusters. Complexity is often non-linear, and such techniques may not be practical for a regular use, but on a single large dataset, they provide a basis by which to evaluate the performance of the genetic algorithm

	Kasumi Cell Line			Yeast Cell Cycle		
	Enumeration	G.A.	P.G.A.	Enumeration	G.A.	P.G.A.
1 condition	-671,040	0%	0%	-428,719	0%	0%
2 conditions	-570,228	0%	0%	-286,690	0%	0%
3 conditions	-431,639	0%	0%	-191,437	0%	0%
4 conditions	-321,680	0%	0%	-150,401	0%	0%
5 conditions	-239,359	0%	0%	-137,360	26%	0%
6 conditions	-187,814	0%	0%	-111,790	26%	0%
7 conditions	-141,700	0%	0%	-89,911	10%	0%
8 conditions	-109,993	0%	0%	-76,047	$< 1\%$	0%
9 conditions	-85,512	0%	0%	-74,763	9%	0%
10 conditions	-65,373	0%	0%	-77,856	20%	0%
11 conditions	N/A	N/A	N/A	-77,651	15%	0%
12 conditions	N/A	N/A	N/A	-73,878	43%	0%
13 conditions	N/A	N/A	N/A	-68,084	39%	0%
14 conditions	N/A	N/A	N/A	-43,350	0%	0%
15 conditions	N/A	N/A	N/A	-27,093	0%	0%
16 conditions	N/A	N/A	N/A	-12,432	0%	0%
17 conditions	N/A	N/A	N/A	-7,665	0%	0%

Table A.4: Validation of the genetic algorithm on small microarray datasets.
Optimal values obtained by enumeration, and gap between these and values obtained by the local genetic algorithm, (G.A.), and the parallel genetic algorithm (P.G.A.).

on large datasets. Here, we use the Lymphoma dataset, which contains 96 conditions.

The heuristic used for these tests has two components: (i) a promotion operator which, given a current solution with k active conditions, finds the best non-active condition to add to the bicluster and obtain a solution with $k + 1$ conditions; (ii) a demotion operator which, given a current solution with k active conditions, finds the active condition to remove from the bicluster with the best effect and obtain a solution with $k - 1$ conditions. Recursively using these two operators, starting with the "empty" solution and "complete" solution respectively, we obtain useful biclusters of all possible sizes.

The profile obtained is compared with the profile generated by a single run of the parallel genetic algorithm in Figure A.7, (p.151). The algorithm performs very well: for most bicluster sizes, (especially over 40 conditions), it finds a solution with an overall weight similar to that of the solution obtained from the heuristic method. However, there are regions

	Local G.A.			Parallel G.A.		
	Gap	St. dev.	Optimal	Gap	St. dev.	Optimal
1 condition	0%	0	100%	0%	0	100%
2 conditions	0%	0	100%	0%	0	100%
3 conditions	0%	0	100%	0%	0	100%
4 conditions	0.1%	0.4	95%	0%	0	100%
5 conditions	18.0%	11.2	25%	0%	0	100%
6 conditions	19.7%	9.5	15%	0%	0	100%
7 conditions	6.8%	5.2	35%	0%	0	100%
8 conditions	0.8%	0.9	30%	0%	0	100%
9 conditions	5.7%	5.0	40%	0%	0	100%
10 conditions	15.4%	6.4	10%	0%	0	100%
11 conditions	12.1%	9.9	35%	0%	0	100%
12 conditions	30.6%	18.9	25%	0%	0	100%
13 conditions	25.3%	19.1	35%	0%	0	100%
14 conditions	0%	0	100%	0%	0	100%
15 conditions	0%	0	100%	0%	0	100%
16 conditions	0%	0	100%	0%	0	100%
17 conditions	0%	0	100%	0%	0	100%

Table A.5: Local vs. parallel genetic algorithm.
Average gap to optimal solution, standard deviation and frequency at which optimal solutions are found. (Results shown for the Yeast Cell Cycle dataset)

where the solutions obtained are poor, (15-35 conditions), and regions where the solutions obtained are better than that of the heuristic method, (8-12 conditions). It must be noted that this profile corresponds to a single run of the algorithm: while the heuristic method always finds the same profile, variations may occur for the genetic algorithm, especially in regions where the algorithm does not reach local optimal solutions.

To improve the overall performance of the algorithm, a fifth evolution operator is added. An existing solution is randomly selected, and a local search performed: we add a condition and remove an active one, (to maintain the bicluster size), as long as the solution can be improved. For small microarray dataset, this operator does not improve the overall performance, as the parallel algorithm was already identifying the optimal solution for each bicluster size. For large datasets, the profile obtained is shown in Figure A.8. The overall performance is significantly improved, and the algorithm outperforms the previous

implementation over the whole search domain. It also compares very well with the heuristic method over the whole search domain, obtaining better solutions for small sizes, (0-12 conditions), and similar solutions elsewhere. Another interesting result is that the profile obtained is largely conserved over multiple runs: 71 biclusters, (out of 96), are obtained at each run and, among the remainder, 18 have a standard deviation smaller than 10% of the average solution obtained. The latter ones correspond to non-optimal low-energy solutions in which the algorithm gets "trapped". Overall, these results suggest that most of the solutions identified are optimal for their respective bicluster size.

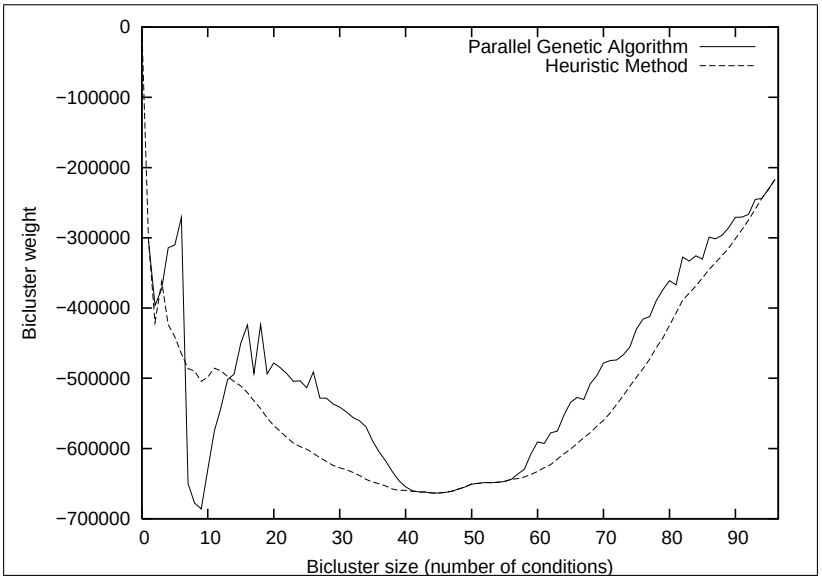


Figure A.7: Solution profile on the Lymphoma dataset.
A single run of the parallel genetic algorithm outperforms the heuristic method used to extract biclusters in some regions, but is less efficient in others.

A.5 Analysis of the overall biclustering technique

A.5.1 Biological significance of the results obtained

We have implemented a biclustering technique consisting of two components which have been assessed and are producing useful results: a weighting scheme, (Section A.3), and a

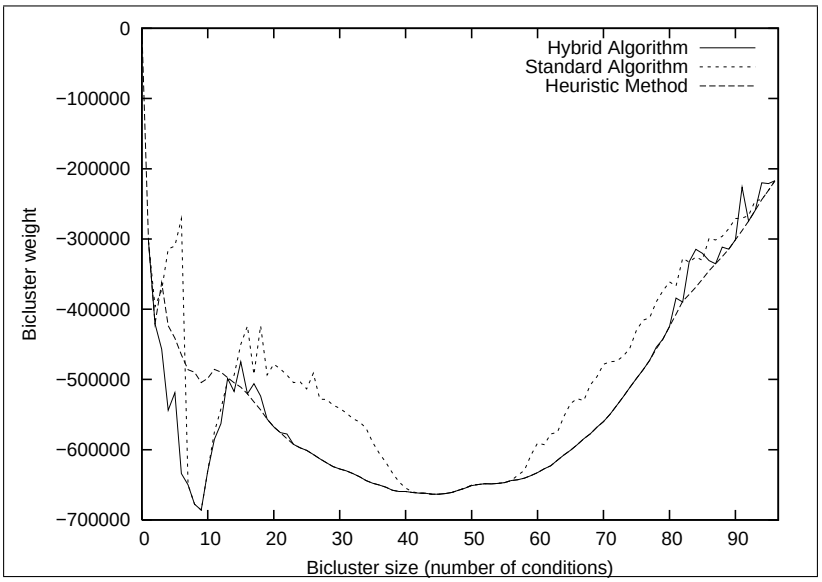


Figure A.8: New solution profile on the Lymphoma dataset.
A single run of the "hybrid" algorithm significantly outperforms the "standard" parallel genetic algorithm overall, and the heuristic method in a specific region , (low bicluster sizes), and obtains results similar to that of this method elsewhere.

parallel genetic algorithm, (Section A.4).

The final step in assessing the biclustering technique is to analyse the biclusters obtained using the overall method. This analysis is performed on the three datasets already used in this Chapter. These datasets have been previously analysed, and this provides a description of function for most of the genes present.

In the Yeast Cell Cycle dataset, the 13-condition bicluster with the smallest overall weight contains two subsets of genes with related functions. The first is related to mitochondrial[8] activity. It contains genes responsible for:

- activation of a mitochondrial acyl carrier protein.
- an ATP-dependent helicase[9].
- a mitochondrial specificity factor which interacts with mitochondrial core polymerase

[8]Mitochondria are membrane-enclosed specialized subunits found in most eukaryotic cells. They produce adenosine triphosphate, (ATP), which the cells use as a source of chemical energy.

[9]Helicases are enzymes which separate two complementary nucleic acid strands.

Rpo41p.

- a subunit of a complex involved in the mitochondrial respiratory chain.

The second subset contains two genes, encoding a protein specifically required for autophagy, and a protein that is a component of autophagosomes[10]. Of the last two genes, one corresponds to a protein of unknown function. The other corresponds to a DNA replication initiation factor. This may be related to helicase activity and to the first subset described.

Next, we consider a 90-condition bicluster obtained from the Lymphoma dataset. A first subset in this bicluster clearly corresponds to immune activity, and contains genes encoding chemokines, interferon[11]-inducible proteins, T cell transcription factors, and carboxypeptidase[12] M. It has been observed that this enzyme is expressed on mature T cells, mainly after activation (de Saint-Vis et al., 1995). Several other genes found in this bicluster have been linked to cancer development, and may be grouped into one subset: phosphatidic acid phosphatase[13] type 2B (Benenson et al., 2004), c-Fos[14] (Prusty and Das, 2005; Shen et al., 2008), and cathepsin B (Yan and Sloane, 2003). The remaining genes have unknown functions.

The results obtained from the Kasumi Cell Line dataset may appear less convincing: with all conditions, the bicluster found contains 88 genes. With fewer conditions, the solutions obtained are even larger. This is not due to the genetic algorithm, since the same solutions are obtained with the exact enumeration method. This is not a consequence of poor weights either, since interesting subsets appear in the biclusters. For instance, in the 88-gene bicluster, we identify all occurences of the human 18S ribosomal RNA gene, which is present in the dataset four times, (full sequence, bases 1-646, bases 647-1292, and bases 1293-1938). We also identify nine proteins which belong to the same family, (associated with brain

[10]Autophagosomes are vesicles which store structures the cell targeted for destruction through autophagy, (a cellular degradation pathway for the removal of damaged, or superfluous, proteins and cell subunits).

[11]Interferons are cytokines. They are produced by the cells of the immune system in response to viruses, parasites or tumor cells.

[12]Carboxypeptidases are enzymes which hydrolyzes the carboxy-terminal, (C-terminal), end of a peptide bond. They have diverse functions, (e.g. catabolism, protein maturation).

[13]This encoded protein is a membrane glycoprotein localized at the cell plasma membrane.

[14]This gene is part of the AP-1 transcription factor, which upregulates a wide range of genes.

cells), and nine others which are associated with modifications of the p53 pathway. The identification of these three meaningful subsets confirms the validity of our approach, and the unusual size of the biclusters obtained is almost certainly a consequence of the configuration of this particular dataset, (more than 20,000 genes, but only ten conditions), which limits discrimination between the genes.

A.5.2 Applications

The analysis technique developed provides useful biclusters on all datasets tested. Such results can be used to better understand gene expression. Several microarray datasets have, for instance, been used to elucidate aspects of cancer initiation, (see e.g. Somasundaram et al. (2002)), identify genes involved in arthritis (Fujikado et al., 2006), or investigate neurological disorders (Greenberg, 2001).

As outlined in Figure A.1, (p.129), these results can also be used to refine the lymph node model, and in particular the agent implementation. Recently, microarrays have been developed specifically for the immune context. These include using a microarray to monitor gene expression in the chicken immune system (Sarson et al., 2007), and analysis of bovine macrophage cells (Jensen et al., 2006). Providing such techniques can be transferred to produce microarrays for the human immune system, the biclustering technique developed and tested in the Chapter would permit extraction of useful immune-specific genetic information which can, in turn, be used to refine the agent implementation in the lymph node model.

Very recently, Shendure (2008) questioned the future of microarray technologies, following promising reports on next-generation sequencing applications (Cloonan et al., 2008; Mortazavi et al., 2008). These techniques are, indeed, an interesting prospect, but we do not think it diminishes the potential of the tools developed here. Irrelevant of the technique employed to obtain them, large datasets always require analysis in order to extract useful information. The technique presented in this Chapter is versatile, and can be adapted to other types of data.

In particular microarray-based techniques are developed for DNA methylation profiling, (see e.g. Schumacher et al. (2006)). This type of data is suitable for analysis based on techniques similar to that developed here, and would be useful for the refinement of the main model layer, as methylation and other epigenetic changes are involved in the immune system. These changes are detailed in Chapter 7.

A.6 Chapter summary

In this Chapter, we introduced a new layer to the immune system model. The motivation for this layer, which investigates gene expression, is that the immune interactions implemented in the lymph node model are, in part, controlled by underlying genes. To simultaneously investigate the expression patterns of multiple genes under several conditions, microarrays datasets are produced, and analysed using techniques such as biclustering. Here, we proposed a new approach to biclustering, which incorporates a novel weighting scheme and a parallel genetic algorithm. Both components are individually evaluated, and their combination is shown to provide biologically meaningful sets of genes.

The method implemented is a useful addition to the main model layer as well as a powerful tool which can be applied to other systems where microarrays are also used, such as identification of genes involved in cancer.

Appendix B

Selected articles

B.1 ERCIM News 64

Perrin, D. (2006). Agent-Based Modelling of Viral Infection. *ERCIM News*, 64:50–51.

This article is a first introduction to the objectives and challenges of the immune system model, and gives an early outline of the modelling process.

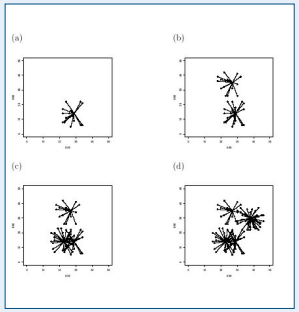


Figure 1: Development of 4-epitope network. Only CTL clonotypes are shown. Centres and edges = immunogenic epitopes and stimulated CTL respectively. By (c) there is a pool of memory CTL clonotypes, some close enough in shape space to exert clearance pressure against the heterologous virus, (non-leaf nodes deg (2)). By (d), suppression of final infection sees cross-reactive contribution of memory cells, specific to a virus, thus resulting in a network connection.

Response: A Shape Space, Physical Space Model: Theor. in Biosci., 123(2):183-194). A network model of shape space emerges naturally from the real-space model, with each immunogenic epitope e_k and activated CTL clonotype c_j represented by a node in the space. Clearance pressure applied between two nodes forms a directed edge between them, with a weight dependent on the affinity. After initial infection, most c_j undergo programmed apoptosis (a crucial regulator of immune system homeostasis), which corresponds to edge deletion. However, recruitment to the memory pool consumes a given percentage of activated CTL, so nodes remain active in shape space, preserving edge connections to stimulatory epitopes.

Recent work indicates that effector CTL memory cells can recognise epitopes of unrelated viruses, so that heterologous viruses (derived from a separate genetic source) may be a key factor in influencing the hierarchy of CD8+ T-cell responses and the shape of memory T-cell pools. Shape space can be used to model both homogeneous viruses with conserved and mutated epitopes, and homologous viruses, with cross-reactive epitopes.

Early and protective immunity can be mediated by memory T-cells generated by previous heterologous infection (represented graphically by nodes with degree > 2). Cluster linkage illustrates conditions by which immunity to one virus can reduce the effects of challenge by another (see Figure 1). Damage to or suppression of critical cross-reactive 'a'-nodes has significantly greater impact than damage to leaf or 'b'-nodes. Different disease outcomes to identical infection strains can be explained in terms of density and distribution of a-nodes. The degree of protection from cross-reacting memory cells depends on the distance between the memory T-cell clonotype and immunogenic epitope, with optimal immunity for re-infection by the same antigenic epitope. Results show increasingly effective clearance dynamics as the memory pool increases and each T-cell clone has a finite chance of becoming a long-lived memory cell.

Please contact:
Heather J. Ruskin
Dublin City University / IUA, Ireland
E-mail: hruskin@computing.dcu.ie

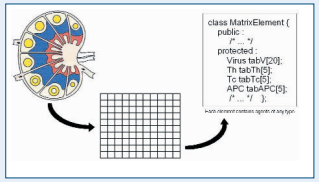
Agent-Based Modelling of Viral Infection

by Dimitri Perrin

The three phases of the macroscopic evolution of the HIV infection are well known, but it is still difficult to understand how the cellular-level interactions come together to create this characteristic pattern and, in particular, why there are such differences in individual responses. An 'agent-based' approach is chosen as a means of inferring high-level behaviour from a small set of interaction rules at the cellular level. Here the emphasis is on cell mobility and viral mutations.

One of the most characteristic aspects of the HIV infection is its evolution: in the initial short acute phase the original viral strains are destroyed, in the second year-long latency period, the number of strains slowly increases and, in the final phase, Acquired ImmunoDeficiency Syndrome (AIDS) develops when the immune system is no longer able to cope with the multiplying strains and is overcome. The principal aim of this work, based at Dublin City University, is to try to understand why the range of experience with respect to HIV infection is so diverse. In particular, the work aims to address questions relating to variation in the length of the individual latency period. This may be very long (for relatively low success of antipathetic mutation) in one individual, compared to another with much higher mutation levels.

The indications are that the observed variation lies in the priming and initial level of fitness of the immune response of the individual, together with the various factors that influence this. If such 'priming patterns' can be recognized or even predicted, then in the long term we may have a way of 'typing' an individual and targeting intervention appropriately. Unfortunately, understanding how the immune system is primed by experience of antigenic invasion and diversity is non-trivial. The challenge is to determine what assumptions can be made about the nature of the experience, and then modelled, tested against clinical data and hence argued plausibly. The aim is to understand how the cell interactions lead to the observed endpoints. What exactly is involved in antigenic diversity? How variable is the mutation rate and the viral load? What is the importance of cell mobility and how realistic is this in terms of cross-infection and subsystem involvement? How important then is the cross-reactivity?



The lymph node (adapted from N. Levy, Pathology of lymph nodes, 1996) is modelled as a matrix in which each element is a physical neighbourhood and can contain several agents of each type.

The immune response is dynamic and includes growth and replenishment of cells and in-built adaptability, through mutation of its defences to meet new threats. It also includes aspects of cell mobility, which may be captured by means of defining the movement and affinity of cell-types in a defined spatial framework. In particular, this will enable us to study the variation in viral load and the way in which the host response may lead to degradation of protection.

To investigate these questions, an 'agent-based' approach is chosen as a means of inferring high-level behaviour from a small set of interaction rules at the cellular level. Such behaviour cannot be extracted analytically from the set of rules, but emerges as a result of stochastic events, which play an important part in the immune response.

The initial model consists of agents (or functional units) with designated properties that mimic the operation of a single lymph node (as a test case). This prototype, however, includes all known interactions contributing to cell-mediated immunity and the local evolution of the virions. The antibody-mediated response has not been considered initially, because the cell-mediated arm plays a dominant role in repelling attack. The agents implemented represent Th (helper) and Tc (cytotoxic) lymphocytes, Antigen Presenting Cells and virions. They inherit from a common C++ class designed to deal with features such as mobility. Each class then implements through attributes and methods the specific properties of each cell type, such as the activation of a Tc cell by a Th cell. The lymph node is modelled as a matrix in which each element is a physical neighbourhood able to contain various agents of each type.

The next step is to increase of the number of lymph nodes. This extension involves millions of agents and requires major computational effort, so that parallelization methods are inevitable. The use of these methods is a natural consequence and advantage of the multi-agent approach. A human body contains hundreds of lymph nodes. The aim here is to extend the size and complexity of the systems that can be modelled to something approaching reality.

The representation of the innate response as a common background and the adaptive part as a characterized set of features will be the next step. This will allow the development of a large system to be studied over a longer period, in order to focus on disease progression endpoints and intervention effects.

The author would like to thank the Irish Research Council for Science, Engineering and Technology for the funding made available through the Embark Initiative.

Please contact:
Dimitri Perrin, Dublin City University/
IUA, Ireland
E-mail: dperrin@computing.dcu.ie

The 'Decent' Project: Decentralized Metaheuristics

by Enrique Alba and Martin Middendorf

The project 'Decent' (Decentralized Metaheuristics) is developing new swarm-based metaheuristics in which a decentralized design leads to emergent phenomena. These are not only important for solving complex problems, but are suitable for parallel execution in computing grids and multi-task hardware platforms.

Metaheuristics, such as evolutionary algorithms (EA), ant colony optimization (ACO), simulated annealing (SA) and particle swarm optimization (PSO), are methods that can be applied successfully to almost all optimization problems. For many hard tasks in a huge variety of areas, these algorithms are considered to be the state-of-the-art methods; these areas include engineering applications, bioinformatics, telecommunications, logistics and business.

Due to their practical importance and the need to solve large real-world instances of hard problems, parallel algorithms have been proposed for most metaheuristics. However, most approaches do not utilize the full potential of parallel execution because of their synchronicity and execution on clusters of homogeneous machines. All this makes it difficult to apply them to interesting parallel systems such as dynamically reconfig-

B.2 LNCS 3980

Perrin, D., Ruskin, H. J., Burns, J. and Crane, M. (2006). An agent-based approach to immune modelling. *Lecture Notes in Computer Science*, 3980:612–621.

In this article, we introduce the overall challenges of immune system modelling, and presente the agent-based lymph node model. We also give a first outline of the lymph network model, and of the requirements for a parallel implementation.

An Agent-Based Approach to Immune Modelling

Dimitri Perrin, Heather J. Ruskin, John Burns, and Martin Crane

Dublin City University, School of Computing, Dublin 9, Ireland
dperrin@computing.dcu.ie

Abstract. This study focuses on trying to understand why the range of experience with respect to HIV infection is so diverse, especially as regards to the latency period. The challenge is to determine what assumptions can be made about the nature of the experience of antigenic invasion and diversity that can be modelled, tested and argued plausibly. To investigate this, an agent-based approach is used to extract high-level behaviour which cannot be described analytically from the set of interaction rules at the cellular level. A prototype model encompasses local variation in baseline properties contributing to the individual disease experience and is included in a network which mimics the chain of lymphatic nodes. Dealing with massively multi-agent systems requires major computational efforts. However, parallelisation methods are a natural consequence and advantage of the multi-agent approach. These are implemented using the MPI library.

Keywords: HIV, immune response, complex system, agent-based, parallelisation methods.

1 Introduction

The objective of this study is to understand why the range of experience with respect to HIV infection is so diverse. In particular, the work aims to address questions relating to variation in length in individual latency period. This may be very long (for relatively low success of antipathetic mutation) in one individual, compared to another with much higher mutation levels.

The indications are that the observed variation lies in the priming and initial level of fitness of the immune response of the individual, together with the various factors influencing this [1]. If such "priming patterns" can be recognised, or even predicted, then in the long term we may have a way of "typing" an individual and targeting intervention appropriately. Unfortunately, understanding how the immune system is primed by experience of antigenic invasion and diversity is non-trivial [1]. The challenge is to determine what assumptions can be made about the nature of the experience, can be modelled, tested against clinical data and hence argued plausibly. The aim is to understand how the cell interactions lead to the observed endpoints.

The immune response is dynamic and includes growth and replenishment of cells and in-built adaptability, through mutation of its defences to meet new threats. It also includes aspects of cell mobility, which may be captured, by means

M. Gavrilova et al. (Eds.): ICCSA 2006, LNCS 3980, pp. 612–621, 2006.

of defining movement and affinity of cell-types in a defined spatial framework. In particular, this will enable study of variation in viral load and the way in which host response may lead to degradation of protection.

To investigate these questions, an "agent-based" approach is chosen, as a means of inferring high-level behaviour from a small set of interaction rules at the cellular level. Such behaviour cannot be extracted analytically from the set of rules [1], but emerges as a result of stochastic events, which play an important part in the immune response [2].

The initial model consists of functional units, called agents, with designated properties which mimic the operation of a single lymph node. This test-case prototype, however, includes all known interactions contributing to cell-mediated immunity and the local evolution of the virions. The antibody-mediated response has not been considered initially, because the cell-mediated arm plays a dominant role in repelling attack. The agents implemented represent Th (helper, or CD4) and Tc (cytotoxic, or CD8) lymphocytes, Antigen Presenting Cells, and virions. The computational structure of the numerical experiments is based on inheritance from a common C++ class designed to deal with features such as the mobility and then each class includes specific attributes and methods to implement specific properties of each cell type. The lymph node itself is modelled as a matrix, in which each element represents the physical neighbourhood of a cell type, (in terms of its agent neighbours). The frequency with which an infected cell will produce a new virion is used as the simulation time-step. At each time step, agents can move from one matrix element to another, and interact with the other agents present in their physical neighbourhood (i.e. with cell types in the same neighbourhood).

Current development is focused on increasing the number of lymph nodes, which involves millions of agents, requiring major computational effort and parallelisation methods. These are, however, a natural consequence and advantage of the multi-agent approach [3]. The aim is to extend the size and complexity of the systems modelled to something approaching realism.

2 A Complex Biological Mechanism

2.1 The Immune Response Against a Viral Attack

Immunity can be defined as all mechanisms which allow the body recognition of that which belongs to its system and consequently tolerate it, and recognise what does not and fight to eradicate it. The immune system is complex and involves various types of cells. When a foreign element is recognised, it can be dealt with in two different ways: the immune response can be non-specific or specific. A non-specific response is based upon the fact that the foreign element does not show, at its surface, the antigens characterising the cells belonging to the body. This is the response that has to be diminished when transplants are carried out. In contrast, the specific response is based on the accurate recognition of foreign antigens. This response can be cell-mediated or antibody-mediated. The second one, also known as humoral response, is carried out by B lymphocytes

and mainly targeted at bacterial attacks. We present here a few details about the cell-mediated response, targeted more specifically at viral attacks and taking place in lymphatic nodes. More details about the immune system can be found in specialised journals and immunology courses, such as [4].

The effector cell, in the cell-mediated response, is the Tc lymphocyte. However, it cannot act on its own, needing a chain reaction to achieve activation The first step is carried out by Antigen Presenting Cells which recognise foreign biological entities and start presenting these antigens at their surface. It will then encounter Th lymphocytes. If a Th cell encounters an APC presenting an antigen, which it has been specifically designed to recognise, it activates itself. The Th cells main function is then to coordinate the immune response by activating specific Tc cells.

2.2 The HIV Expansion Strategy

HIV virions use the Th cells described above as hosts to multiply themselves, as detailed in [5]. The gp120 glycoprotein of the virion envelope first attaches itself to the CD4 receptor, characteristic of these immune cells. Then the virion fuses with the lymphocyte using gp41 and the viral RNA is freed into the cell. The viral reverse transcriptase copies the RNA into DNA and integrates it into the cellular DNA. To be successful, this integration has to take place in activated cells. More details about this process can be found in [6]. An important aspect is the high rate of mutation: there is on average a transcription error every 10.000 nucleotides. Since the HIV genome contains about 10.000 nucleotides, this means there is on average a single difference between two "brother virions". All these mutants of course have various fates. On the one hand, most of them will result, for instance, in the suppression of an enzyme, and will be unsuccessful. On the other hand, a mutation can be successful and, for instance, modify the envelope glycoprotein, thus allowing the new virion to temporarily escape from the immune system.

The macroscopic evolution of the disease is divided into three phases. The first one corresponds to the typical immune response against a viral attack. The production of lymphocytes specific to the viral strains is launched, and within a few weeks, all the original strains are eradicated. The mutation rate here becomes critical. It has allowed the appearance of new strains, which have not been detected by the organism yet, and can therefore develop freely. As soon as a strain becomes too intrusive, its detection probability increases and it is eradicated. During this second phase, there are no symptoms. This is known as the latency period, and can last up to ten years. The immune system is heavily loaded, and the destruction of each strain also implies the destruction of the infected cell. A time comes when the immune system cannot cope with the ever increasing number of strains or remain viable, given a strong decrease of the number of the Th cells. During this last phase, known as AIDS (acquired immunodeficiency syndrome), the whole immune system is diminished and opportunistic diseases start appearing, leading to the death of the patient.

3 Simple Rules to Control the Agents

3.1 The Agent-Based Approach

There is no unique definition of what an agent is. However, Wooldridge and Jennings proposed in [7] a definition which is widely accepted and specifies characteristics that an agent must have. An agent has to be autonomous: it can act without any intervention and has some control over its actions and its internal state. It has a social behaviour: it can interact with other agents thanks to a specific language. It can also react: the agent has the ability to scan part of its environment and change its behaviour to take advantage of it. The agent is proactive: it not only reacts to its environment but also acts and takes initiatives so as to satisfy goals. Building on this definition an agent-based model is a model in which the key abstraction elements are agents.

Obviously, each agent has only a limited knowledge of the world in which it evolves, and communication between agents is therefore an important aspect of this approach. This communication is sometimes referred to as linguistic actions, as opposed to non-linguistic actions which are modifications of the environment. Interaction between agents is not limited to communication: they have to share their environment. This implies that agents' actions have to be coordinated. Of course coordination does not mean cooperation: a good competitor maximizes his advantage by coordinating his actions according to the others' decisions. It also does not imply reciprocity of action: a car driver can go past another and coordinate this safely without the second driver knowing it. The key factor when choosing a coordination strategy is the size of the agent population. If every agent can interact with every other one, the number of interaction pairs increases quadratically with the population size. If interaction can occur between several agents instead of pairs, the coordination overhead increases exponentially and can easily exceed the computing facilities [8]. Developing a coordination strategy is therefore both essential and difficult. In many cases, managing to avoid conflicts and blocks is itself an important achievement. This gives us the opportunity to put the emphasis on the main drawback of this approach: it is highly resource-consuming. However, the approach also provides a solution as it is often combined with parallel methods. We develop this idea later on (section 4.2).

This approach being generic, it has been used in various fields. It has for instance been used for aerial traffic planning [9], vehicle monitoring [10] and even to manage chirurgical intensive care units [11]. It has also been extensively used in Natural Sciences, as it provides a very intuitive way to model systems: biological entities are implemented as agents, and interactions between them are dealt with through linguistic and non-linguistic actions among the agent population. In particular, the immune system itself is a discrete system in which the individual behaviour of every cell adds to create to high-level behaviour of the whole system. A simple set of local rules can therefore provide an accurate model of this complex system. This is the approach we have chosen to take.

As we have seen earlier, most of the immune response against HIV is taking place in the lymphatic nodes. The world we model need only be a network of

such nodes. The communication inside the network will be discussed later (section 4.1). Each node is implemented as a matrix. Each element of the matrix correspond to a physical neighbourhood. All the interactions between the agents therefore happen inside this local element and there is no need to consider surrounding matrix elements as would be done if using Moore or Von Neumann neighbourhoods [12].

3.2 The Implemented Features

There are several platforms supporting generic agent-based environments, such as Swarm [13]. However, due to the high number of agents we plan to simulate, we think it is more efficient to have an approach fully dedicated to this particular environment, and therefore optimized. Because of the very detailled knowledge of the cell interactions, we are using a bottom-up approach: we first specify in detail the individual parts of the system (here, the agents), we then link them together to form layer componants (here, the lymphatic node), which are in turn linked until a complete system is formed (here, the lymphatic network).

As noted earlier, this study focuses on the cell-mediated response. Thus, we first need to implement three types of cells, corresponding in the code to three types of agents: Th and Tc lymphocytes, and Antigen Presenting Cells (APC). Of course, we also need a fourth type of agent to model the virions. Each type is implemented into the code using a specific C++ class.

Interestingly, even if all four types of cells have totally different roles, they have a common feature that we want to take into account, i.e. their mobility. This is implemented by another class. This class is then inherited by the four types described above. It also implements other basic properties such as the age of the agents and allow us to have the four agent classes contain only specific features; an advantage of object-oriented programming.

An agent coding a virion only has one specific attribute in the model, its viral strain. In order to prevent the code from allocating too much memory for each agent, the viral strain is only coded as an integer which links to the corresponding strain in an array containing all the useful properties of the strain (e.g. lymphocytes which recognize it, immunogenicity, etc.). The agent has a short-term and partial knowledge of its environment. It is partial in the sense that it is only knows whether there are Th cells in its physical neighbourhood (i.e. the matrix element). It is short-term in the sense that it has no memory of the evolution of the number of lymphocytes. This knowledge is the only piece of information it needs, since its unique objective is to infect a Th cell. Therefore, the typical behaviour of a virion in the model can be given as the following triptych, repeated until a lymphocyte is infected: the agent moves, scans its environment looking for a Th cell, and if possible infects the immune cell.

A Th agent has three specific attributes in the model: an integer coding its surface antigens, another integer coding its "activation state" and a third integer coding its "infection state". Once again, it has no memory of its environment and the only part it knows of it is reduced to the presence, or not, of Tc agents. If the agent is neither activated nor infected, both integers coding the states are

set to zero, and the agent's objective is only to be ready to answer an attack. There is therefore no particular action, apart from moving. The objective of an activated agent is to activate Tc cells. Its "activation state" is set to the value coding the viral strains which activated it, so that it can communicate on the threat. If the agent is infected, it produces new virions belonging to the strain coded in its "infected state", or to a new one if there is a mutation.

A Tc agent has four specific attributes: its surface antigens, its "activation state", its "expansion state" and its "memory state", all implemented as integers. The Tc agents also have a short-term and partial view of their environment: each looks only for agents having the antigens corresponding to the strain which activated it, and destroys them. When activated, an agent multiplies itself during an expansion phase, corresponding to a non-zero "expansion state". After an immune response, a small amount of the Tc agents will become memory cells: their "memory state" will keep track of the strain they fought, the reactivation will be easier, and if reactivated, the expansion phase will be more productive.

An APC agent only has one specific attribute, its "presenting state", coded as an integer. As long as the agent is not presenting any antigen at its surface, the integer stays at zero, and the agent's behaviour is focused on moving and looking for "foreign" entities in its physical neighbourhood, in order to get antigens to present. Then, the "presenting state" codes the strain corresponding to the antigens, and the agents starts looking for Th agents in order to activate them, if they are geared recognise this particular antigen.

Another aspect of the implementation chosen is the allocation of the agents. Memory allocations are among the slowest operations on a computer, and here, we have a model in which thousands of agents are created and destroyed every iteration. Dynamic allocations would make the program too slow. The approach we have chosen is to have, in each matrix element, a static allocation of the maximum number of agents we want to implement. Then, an agent moving from an element to another is coded as the transfer of its attributes from one static memory slot to another. Every agent being small and with few attributes, this gives satisfying results.

3.3 How to Deal with Stochastic Events?

In this model, most methods and functions have to include random number generation. This is due to the fact that many aspects of the real-life system involve stochastic events. More details can be found in [2], but here are a few examples. First, an aspect we have to deal with is the process by which new lymphocytes are created. A lymphocyte can only recognize a specific set of antigens so, to protect itself against any attack, the body has to generate thousands of "variations" between lymphocytes. This has to be implemented using random numbers. Likewise, we noted that one of the most decisive features of the virions is their high mutation rate, and this implies another use of random numbers. Finally, there is no sensible way to deal with mobility unless we include stochasticity.

Stochastic events are essential to this work and a reliable random number generator is needed. A full-scale model will involve millions of agents in very

long simulations. Therefore, the generator also has to be very efficient. As parallel aspects are involved, it would also be a plus for the generator to include such features. There are many generators available, and good ones can also be designed explicitely (see e.g. [14]). However, due to our model requirements, what is needed here is a top-quality parallel generator, and we chose to use the Scalable Parallel Random Number Generators library (SPRNG) [15]. This library incorporates recent, state-of-the-art, developments in the mathematics and computer science of parallel pseudorandom number generation. It an efficient library with an existing, active, user base, ensuring high standards. It allows the streams to be also absolutely reproduced, for computational verification, independent of the number of processors used in the computation and of the loading produced by sharing of the parallel computer. Using it, we can be confident we will produce statistically significant results at a very low computing cost.

4 Interactions Between the Lymphatic Nodes

4.1 Sharing Knowledge and Transferring Agents

The immune system is organised so that every lymphatic node is a small defence unit in which the immune response is taking place. There is no need for the response to take place in every node, which is why we built our model as a network of independent matrices (putting the emphasis on the local model of the node). The only physical exchange between lymphatic nodes happens through the recirculation and the mobility of cells which go from one node to another. Each node in the model therefore needs an entry point and an exit point. If, when moving inside the node, an agent reaches the exit point, it is removed from the node and put into a transfer list. The list is dealt with at the end of the iteration. In the meantime, other agents move, interactions take place, as time passes. This accounts for the time it takes the agent in real-life to commute between two nodes. The way in which agents are transferred between the nodes mimics the transfer between matrix elements: we consider only attributes, rather than the agent itself. Thus, an entry in the transfer list contains the type of the agent, its attributes, and its destination. At the end of the iteration, all lists are put together and the moving agents are transferred to the entry point of their destination node.

The other aspect of the communication between our nodes is inherent to our implementation. Since we decided not to put all the strain properties into each agent, we need a way to code them somewhere and make them available to all the agents, wherever they are in the model. These are important properties, and must not be neglected. For instance we need to know, for each strain, which lymphocytes will recognise it for sure and which lymphocytes might recognise it. One characteristic is that when a lymphocyte from the second category recognise the strain, it moves from the second into the first. This is critical to the realism of the model, since it allows us to introduce some adaptability and emergent behaviour. One answer could have been to create a linked list containing the strains active in the current simulation. The obvious advantage is to limit the size allocated to the strains to what is actually needed. However, it has one major drawback which makes it pointless in our case, namely that the high mutation rate means a large number of strains, increasing as the simulation continues. The bigger the list, the longer it will take to get the properties for a particular strain and since this list has to be accessed thousands of times in every iteration, this process would slow the whole program down. We therefore decided to have an array of strains. This array is large (i.e. tens of thousands of strains) and represents potential strains for the simulation to be implemented. Considering that a strain in the array can account for various strains in real life (if they differ on properties we do not code explicitely), we are confident this should give us enough diversity.

4.2 Parallelisation Efforts

When the program is running at full scale, each node contains hundreds of thousands of agents. In real life, a human body contains about a thousand lymphatic nodes. Matching this value is a long-term objective and may not be achievable, but even with fifty nodes, we would have to deal with millions of agents. The time-step of the program is about fifty seconds, so about six million iterations are needed for a 10-year simulation. Running such a program on a single computer would take months, and not even have enough memory might be available to initialize all the matrices. If we also consider the fact that we have to run several simulations to statistically assess the role of each parameter such as the mutation rate, a parallel approach makes even more sense.

The approach we develop here is to mimic the immune system, in the sense that each lymphatic node will be computed by a different computer (also called node) on a cluster. As the lymphatic nodes are mainly independent from each other, this is the best way to take advantage of the parallel option. Moreover, the local model is already known to run on a single computer so approximate expectations on performances are known also. This type of spatial parallelisation has been studied in [16] for Monte-Carlo simulations. The main disadvantage in that study is the communication overload. Here, most of the communication taking place on the cluster is the transfer of agents from one node to another. Using the list process described above, this is kept to a minimum. This parallel approach is implemented using the Message-Passing Interface (MPI) [17, 18]. It is under validation on a cluster composed of a Dell PowerEdge 1750 acting as the master node and sixteen of these machines acting as slaves. More important clusters will also be used for full-scale runs.

The most difficult part here is to deal with the updates of the array containing the strain. On the one hand, if we keep only one array (on the main node of the cluster) it would lead to excessive communication: each agent would have to ask for the viral strain properties at each iteration. On the other hand, having an array linked to every node would impose a process to make sure that at every instant all arrays contain the same information, for all the strains. Using MPI advanced features, this can be done through “collective communication”.

This approach provides an intuitive way to combine the parallel computing features with a process which mimics the immune system. The transfer of the agents is currently being optimized. This will allow us to then run full-scale simulations. Our objective is to first reproduce the three-phase evolution of the disease and then alter the parameters (mobility, viral load) to study how they affect the latency period length.

5 Conclusion

The objective of this study is to understand why the range of experience with respect to HIV infection is so diverse, addressing in particular questions relating to variation in length in individual latency period. To investigate these questions, an "agent-based" approach is chosen, as a means of inferring high-level behaviour from a small set of interaction rules at the cellular level as well as including stochastic events.

The model developed mimic the immune system, as it is organised as a network of matrices, each of them corresponding to a lymphatic node. Matrix elements can host several agents, of four different types, accounting for virions, Th and Tc lymphocytes, and Antigen Presenting Cells. Thus, it is possible to model the HIV spreading strategy and the cell-mediated immune response.

Because the system we study is so complex, millions of agents are needed, and it is not possible to run the model on a single computer. Therefore, parallel methods are implemented. Using MPI, every lymphatic node is allocated to a different computer on a cluster, and "collective communication" is used to share knowledge common to all nodes.

This parallel implementation is currently being tested and the first results should be available in the coming months.

Acknowledgements

The authors would like to thank the Irish Research Council for Science, Engineering and Technology for the funding made available through the Embark Initiative.

References

1. Burns, J.: Emergent networks in immune system shape space. PhD thesis, Dublin City University, School of Computing, 2005.
2. Germain, R.N.: The Art of the Probable: System Control in the Adaptive Immune System. Science 239 **5528** (2001) 240–245.
3. Jennings, N., Sycara, K., Wooldridge, M.: A roadmap of agent research and development. Autonomous agents and multi-agents systems 1 **1** (1998) 7–38.
4. Lemahieu, J.C..: Le systeme immunitaire. Immunology courses [French] (available online at http://anne.decoster.free.fr/immuno/orgcelri/orgcelmo.htm), last access on December 14th, 2005.

5. Klatzmann, D., Champagne, E., Chamaret, S., Gruest, J., Guetard, D., Hercend, T., Gluckman, J.C., Montagnier, L.: T-lymphocyte T4 molecule behaves as the receptor for human retrovirus LAV. Nature 312 **5596** (1984) 767–768.
6. Decoster, A., Lemahieu, J.C..: Les retrovirus. Immunology courses [French] (available online at http://anne.decoster.free.fr/d1viro/vretrov0.html), last access on December 14th, 2005.
7. Wooldridge, M., Jennings, N.: Intelligent agents: Theory and practice. The Knowledge Engineering Review 2 **10** (1995) 115–152.
8. Durfee, E.H.: Scaling up agent coordination strategies. Computer 34 **7** (2001) 39–46.
9. Cammarata, S., McArthur, D., Steeb, R.: Strategies of cooperation in distributed problem solving. roceedings of the Eighth International Joint Conference on Artificial Intelligence (IJCAI-83), Karlsruhe, Germany (1983).
10. Durfee, E.H.: Coordination of distributed problem solvers. Kluwer Academic Publishers (1998).
11. Hayes-Roth, B., Hewett, M., Washington, R., Hewett, R., Seiver, A.: Distributing intelligence within an individual. Distributed Artificial Intelligence Volume II, L. Gasser and M. Huhns (editors), Pitman Publishing and Morgan Kaufmann (1989) 385–412.
12. Kari, J.: Theory of cellular automata: A survey. Theoretical Computer Science 334 **2005** (2005) 3–35.
13. Minar, N., Burkhart, R., Langton, C., Askenazi, M.: The Swarm simulation system: A toolkit for building multi-agent simulations. Working Paper 96-06-042, Santa Fe Institute (1996).
14. Press, W.H., Vetterling, W.T., Teukolsky, S.A., Flannery, B.P.: Numerical Recipes in C++: the art of scientific computing. Cambridge University Press (2002).
15. Srinivasan, A., Mascagni, M., Ceperley, D.: Testing parallel random number generators. Parallel Computing 29 **2003** (2003) 69-–94.
16. Hecquet, D., Ruskin, H.J., Crane, M.: Optimisation and parallelisation strategies for Monte Carlo simulation of HIV infection. Submitted to Computers in Biology and Medicine (2005).
17. Gropp, W., Lusk, E., Skjellum, A.: Using MPI: Portable Parallel Programming With the Message-Passing Interface, second edition. MIT Press (1999).
18. Gropp, W., Lusk, E., Skjellum, A.: Using MPI-2: Advanced Features of the Message Passing Interface. MIT Press (1999).

B.3 ERCIM News 72

Perrin, D., Ruskin, H. J., Crane, M., and Walshe, R. (2008). Epigenetic modelling. *ERCIM News*, 72:46.

This article introduces to a non-specialist audience, (mathematicians and computer scientists), the field of Epigenetics, and presents our overall research project. The model presented in this Thesis is a proof of concept for this long-term study.

Epigenetic Modelling

by Dimitri Perrin, Heather J. Ruskin, Martin Crane and Ray Walshe

The field of epigenetics looks at changes in the chromosomal structure that affect gene expression without altering DNA sequence. A large-scale modelling project to better understand these mechanisms is gaining momentum.

Early advances in genetics led to the all-genetic paradigm: phenotype (an organism's characteristics/behaviour) is determined by genotype (its genetic make-up). This was later amended and expressed by the well-known formula $P = G + E$, encompassing the notion that the visible characteristics of a living organism (the phenotype, P) is a combination of hereditary genetic factors (the genotype, G) and environmental factors (E). However, this method fails to explain why in diseases such as schizophrenia we still observe differences between identical twins. Furthermore, the identification of environmental factors (such as smoking and air quality for lung cancer) is relatively rare. The formula also fails to explain cell differentiation from a single fertilized cell.

In the wake of early work by Waddington, more recent results have emphasized that the expression of the genotype can be altered without any change in the DNA sequence. This phenomenon has been tagged as epigenetics. To form the chromosome, DNA strands roll over nucleosomes, which are a cluster of nine proteins (histones), as detailed in Figure 1. Epigenetic mechanisms involve inherited alterations in these two structures, eg through attachment of a functional group to the amino acids (methyl, acetyl and phosphate). These 'stable alterations' arise during development and cell proliferation and persist through cell division. While information within the genetic material is not changed, instructions for its assembly and interpretation may be. Modelling this new paradigm, $P = G + E + EpiG$, is the object of our study.

To our knowledge, no previous efforts have sought to model directly the mechanisms that affect epigenetic changes. Biological research on epigenetic phenomena is ongoing, but while some very promising articles are being published, most still contain only qualitative descriptions of epigenetic changes. This is not ideal when trying to develop computer-based models, but it is also not unusual. Over a decade ago the basics of HIV infection were understood, but quantitative data were sparse. Yet as early as 1992, differential equation models were proposed, while cell-mediated micro-models date from the 1990s. As more data have become available, these models have improved in sophistication, incorporating features such as shape-space formalism and massively multi-agent, parallel systems.

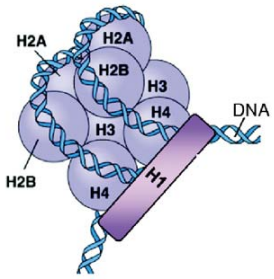


Figure 1: A nucleosome, the fundamental subunit of the chromosome (adapted from C. Brenner, PhD thesis, Université Libre de Bruxelles, 2005).

As a first step, we propose a microscopic model for chromatin structures. From the current biological results, it clearly appears that each unit (eg histone, DNA strand or amino acid) has a distinct role in epigenetic changes, and this role can alter depending on the type or location of the unit (eg which particular amino acid, what part of the DNA strand etc). For efficiency, this is best modelled using an object-oriented approach and a C++ implementation. The main objective of this early model is to provide a description and hierarchy for epigenetic changes at the cell level, as well as an investigation into the dynamics and time scales of the changes. These results will then be used to 'feed into' other models. Already in development are approaches such as agent-based modelling of cell differentiation and complex recurrent networks of cancer initiation by epigenetic changes.

Another early model uses Probabilistic Bayesian Networks. These represent a set of variables and their probabilistic dependencies and are constructed as directed acyclic graphs, for which nodes represent variables and arcs encode conditional dependencies between the variables. The variables can be of any type, ie a measured parameter, a latent variable or even a hypothesis. These networks can be used for inference, parameter estimation and refinement, and structure learning. This approach has been successfully used in medicine (eg breast cancer diagnosis) and biology (eg protein structure prediction), and epigenetic mechanisms appear amenable to such techniques.

Though still in its infancy, the project is gaining momentum and early work on the different approaches looks very promising. Active involvement from biologists and medical researchers is currently being sought in order to secure access to data and guarantee model realism (as highlighted by a presentation at the International Agency for Research on Cancer in early December 2007). Previous modelling experience from the group promises sensible integration of the various approaches and efficient implementations. Several publications and presentations are expected in the coming year, all of which will appear on the group's Web site (link below).

Links:
http://www.computing.dcu.ie/~dperrin/
http://www.computing.dcu.ie/~msc/publications.shtml

Please contact:
Dimitri Perrin
School of Computing, Dublin City University, Ireland
Tel: +353 1 700 8449
E-mail:
dimitri.perrin@computing.dcu.ie

B.4 ERCIM News 74

Perrin, D., and Burns, J. (2008). Large-scale immune models and visualization. *ERCIM News*, 74:33–34.

This article summarises the development of our large-scale lymph network model, and introduces the current collaborative efforts to develop visualisation tools for such simulations.

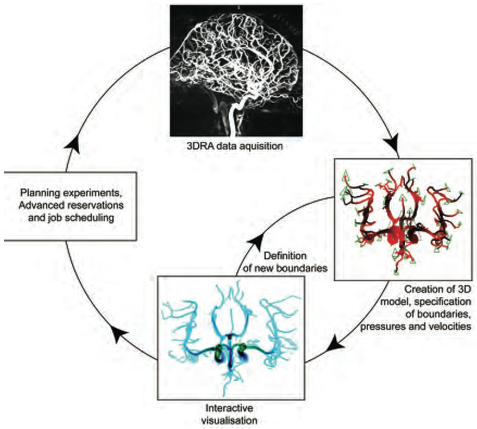


Figure 2: Workflow diagram of GENIUS clinical 'scenario'.

resource provision strategies, at national and international levels, to allocate appropriately scaled resources to projects when they are funded. Such projects often require access to a wide range of resources as part of a scientific workflow, for example the high-end machines provided by DEISA (Distributed European Infrastructure for Supercomputing Applications) or the TeraGrid, along with smaller clusters provided by the UK National Grid Service.

The GENIUS project will be complete in December 2009, at which time a unique test case for the use of patient-specific simulation and high-performance computing resources to plan surgical intervention will be available. As described, however, many challenges remain to be resolved before supercomputing at work becomes the premise of the medical doctor.

Links:
GENIUS project Wiki: http://wiki.realitygrid.org/wiki/GENIUS
VPH Initiative and VPH NoE: http://www.biomedtown.org/biomed_town/vphn
MPICH-G2: http://www3.niu.edu/mpi/ (there is no MPIg link at the moment)
SPRUCE: http://spruce.teragrid.org
Urgent Computing, CTWatch: http://www.ctwatch.org/quarterly/archives/march-2008

Please contact:
Peter Coveney
Centre for Computational Science, Department of Chemistry, University College London, UK
Tel: +44 20 7679 4560
E-mail: p.v.coveney@ucl.ac.uk

Large-Scale Immune Models and Visualization

by Dimitri Perrin and John Burns

Large-scale computing architectures allow detailed modelling of complex systems. Here we present applications in the field of computational biology.

In silico simulation methods – simulations within computer software – have become indispensable tools in the development of expensive new technology, from aircraft manufacture to nuclear power station development. For various reasons however, modelling and simulation techniques have only very slowly been adopted by the biological research community.

Biological systems are typically complex and adaptive. Given the dynamic nature of these phenomena, it is non-trivial to provide a comprehensive description of such systems and, in particular, to define the importance and contribution of low-level unsupervised interactions to the overall evolution process. An agent-based approach is presented here.

The immune response, in particular, consists of relatively simple cell-level interactions, but is capable of highly diverse and individual behaviour, with outcomes that are self-evidently more than the sum of their parts. In what follows, a one-to-one reciprocity between autonomous agents and immune cells is proposed. The emergence of tissue-level and body-level patterns is clearly dependent on the relative sizes of cell populations and the balance these achieve. This is demonstrated, for example, by the course of HIV progression, where the whole immune system collapses once immune cell counts decrease below critical levels.

High temporal granularity is necessary to realistically account for cell mobility and interactions. With current computing resources, this prevents models accounting for every immune cell of every type over the whole body. A compromise between agent diversity and agent population size is therefore required. We have implemented a flexible model, which currently includes viral agents and an agent type for each entity of the cell-mediated response. This guarantees enough diversity to account for principal immune and viral mechanisms, while permitting the simulation of large populations.

In this way, cell mobility can be examined. Agents are explicitly located in structures modelling lymph nodes. Agent movement within these leads to the occurrence of cell-level interactions, and their passage from one node to another. Thus, lymphatic chain structures allow for the spread of viral infection and connected immune response. The model successfully accounts for rapid spreading throughout the initial lymph chain (ie within hours), and slower progression through the whole body (weeks being necessary to obtain infection of all nodes).

Using parallel implementation on a large cluster computer, simulations can model up to a thousand lymph nodes for a total of more than one billion agents. This permits accurate modelling of the immune response to infections, and inclusion of localized effects such as HIV early infection within the gastrointestinal tract, whose importance was recently highlighted. Equivalent large-scale frameworks are also applicable to other complex systems, biological or otherwise. In particular, in silico techniques have enormous potential to contribute to the quality and cost-effectiveness of the drug development process.

Visualization of the results from such simulations is non-trivial. In what follows, we present some initial results in the visualization of immune responses that allows the user to parameterize their in silico experiments in order to study the maximum infection clearance dynamics. The tool also proves to be a useful aid in the classroom for students in their early undergraduate training.

Collaboration between ITT Dublin and Dublin City University has led to the development of novel visualization techniques to study cellular interaction from both the large-scale spatial and temporal perspectives. Visualization of processes offers the researcher the ability to speed up, slow down and hypothesize various parameters. Visualization aids understanding as we rely on visual perception to make crucial decisions. For example, with our initial model, we can visualize the dynamics of an idealized lymphatic compartment, with APC and CD8 cells.

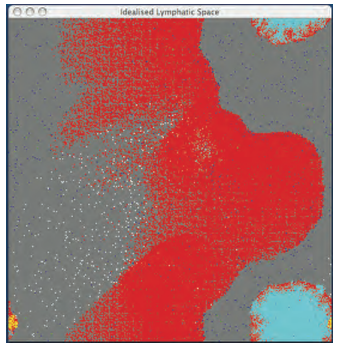


Figure 1: Idealised lymphatic space.

We can parameterize many aspects, such as initial cell levels, rates of change in the life cycle, frequency of infection events and many others. Some viral pathogens are capable of persistent re-infection in that, although population levels of infected antigen presentation cells may decline in response to clearance pressure by a specific CD8 response, over time the number of infected cells rises to chronic and sometimes acute levels. Examples of such viruses are HIV, Human T-lymphotropic virus (HTLV) hepatitis C (HCV), hepatitis B, cytomegalovirus CMV, Epstein-Barr virus (EBV) and rubella.

From the figure we see that as the primary response continues, the effector cells now begin their exit from the lymphatic compartment, after which they will be carried through the blood system and will migrate towards the site of the initial infection (if one exists).

We have developed a 'front end' visualization component to allow students and lecturers in the classroom and lab to experiment with a variety of parameters. We are strongly motivated in this research by the findings of a recent EU report indicating that insilico modelling and simulation of biological processes is a key requirement in the development of cost-effective and timely new disease therapies.

Link:
http://sci-sym.computing.dcu.ie

Please contact:
Dimitri Perrin
Dublin City University
Tel: +353 1 700 8449
E-mail: dperrin@computing.dcu.ie

John Burns
Institute of Technology Tallaght, Ireland
Tel: +353 1 404 2766
E-mail: john.burns@ittdublin.ie

Appendix C

Abstracts

C.1 CGCS '06

Perrin, D., Ruskin, H. J., and Crane, M. (2006). HIV Modelling – Parallel Implementation Strategies. Third International Conference on Cluster and Grid Computing Systems (CGCS'06), Venice, Italy.

Abtract:

We report on the development of a model to understand why the range of experience with respect to HIV infection is so diverse, especially with respect to the latency period. To investigate this, an agent-based approach is used to extract highlevel behaviour which cannot be described analytically from the set of interaction rules at the cellular level. A network of independent matrices mimics the chain of lymph nodes. Dealing with massively multi-agent systems requires major computational effort. However, parallelisation methods are a natural consequence and advantage of the multi-agent approach and, using the MPI library, are here implemented, tested and optimized. Our current focus is on the various implementations of the data transfer across the network. Three communications strategies are proposed and tested, showing that the most efficient approach is communication based on the natural lymph-network connectivity.

Keywords: HIV, Immune modelling, MPI, Parallelisation.

C.2 ICMS '06

Perrin, D., Ruskin, H. J., and Crane, M. (2006). An agent-based approach to immune modelling, Priming individual response. Third International Conference on Modeling and Simulation (ICMS'06), Cairo, Egypt.

Abtract:

This study focuses on examining why the range of experience with respect to HIV infection is so diverse, especially in regard to the latency period. An agent-based approach in modelling the infection is used to extract high-level behaviour which cannot be obtained analytically from the set of interaction rules at the cellular level. A prototype model encompasses local variation in baseline properties, contributing to the individual disease experience, and is included in a network which mimics the chain of lymph nodes. The model also accounts for stochastic events such as viral mutations. The size and complexity of the model require major computational effort and parallelisation methods are used.

Keywords: HIV, Immune modelling, Agent-based system, Individual response.

C.3 ICCM 2007

Perrin, D., Duhamel, C., Ruskin, H. J., and Crane, M. (2007). Microarray biclustering: mathematical model and metaheuristic alternatives. International Conference on Computational Methods (ICCM2007), Hiroshima, Japan.

Abtract:

DNA microarrays are extensively used as a means to obtain expression levels of several genes under a set of conditions, see e.g. [1]. Typically, thousands of genes are considered under tens of conditions. Biclustering is then applied to extract a subset of genes and a

subset of conditions for which it is possible to identify common behaviour [2]. Here, we transform the biclustering into a graph optimisation problem: the microarray is represented as a bipartite graph in which we look for relevant subgraphs. We propose a mathematical formulation for this problem. For the optimisation to give meaningful results, we also introduce a new weighting function, aimed at isolating relevant gene-condition couples. These weights are based on a categorization of the expression ratio. An exact enumerative method is developed and tested on medium-sized arrays (i.e. less than ten conditions) and returns the best solution within a few seconds. Metaheuristics are also implemented, for bigger datasets. Results are again very satisfying and the computation time remains sensible.

Keywords: DNA microarray, biclustering, mathematical formulation, metaheuristics.

References:

[1] F. Oana, T. Homma, H. Takeda, A. Matsuzawa, S. Akahane, M. Isaji, M. Akahane: DNA microarray analysis of white adipose tissue from obese (fa/fa) Zucker rats treated with a 3-adrenoceptor agonist, KTO-7924, Pharmacological Research, Vol.52, pp.395-400, 2005.

[2] H. Turner, T. Bailey, W. Krzanowski: Improved biclustering of microarray data demonstrated through systematic performance tests, Computational Statistics and Data Analysis, Vol.48, pp. 235-254, 2005.

C.4 ICG 2008

Perrin, D., and Ruskin, H. J. (2008). The case for epigenetic modelling (poster). XX International Congress of Genetics, Berlin, Germany.

Abtract:

Objectives. Advances in Genetics incorporate the all-genetic paradigm through the notion that phenotype (P) of an organism is a combination of its genotype (G) and environmental factors (E), i.e. $P = G + E$. More recent work has emphasised "stable alterations" of the chromatin, arising during development and cell proliferation, and persisting through

cell division. While information within the genetic material remains unchanged, instructions for its assembly and interpretation may be modified. Modelling this new paradigm, $P = G + E + EpiG$, is the object of our study.

Methods. To our knowledge, no previous efforts have sought to model directly epigenetic mechanisms. Research on epigenetic phenomena is ongoing, but while promising advances are being reported, most still contain only qualitative descriptions of epigenetic changes. This is not ideal when trying to develop computer-based models, but is also not unusual. A similar situation in the early 1990s necessitated phenomenological development of computer-based models of HIV infection. As more data have become available, these models have improved in sophistication, incorporating new features, and are now a valuable tool for ongoing biological research. Here, we present a development framework based on this experience. Models implemented include microscopic representations and high-level network-based approaches.

Results. Current status of the approaches offers a intuitive and comprehensive representation of epigenetic mechanisms. Efforts at the microscopic level are targeted towards establishing a hierarchy of epigenetic changes, while network-based approaches provide the basis for better development of biomarkers for early detection of cancer.

Conclusion. Though still in its infancy, the project is gaining momentum and early work on the different approaches is encouraging. This establishes a new field, computer-based epigenetic modelling, which is expected to provide valuable insight on key biological questions.

Appendix D

Source Code

The accompanying CD contains commented source code for the three model layers. This code was tested on several platforms. Apart from the MPI library, it does not require installation of extra software.
Unless stated otherwise, all files are written by Dimitri Perrin.

D.1 Agent-based lymph network model

All files required for the main model layer are located in the "HIV" folder. For random number generation, we provide an interface with Mersenne Twister, and this generator is included in the folder, ("mersenne-twister.h").
This is to avoid requirements for SPRNG installation.

D.2 Microarray biclustering

D.2.1 Weighting scheme

All files required for weights generation are available in the "Weights" folder, in "Microarray". As an exemple, we also provide a dataset ready to use, ("MIT.txt"), and the corresponding weights obtained ("weights-MIT.txt").

In this folder, the file “main.c” is the only one directly referring to the work implemented in this Thesis. The other files exclusively deal with data storage, through a previously existing matrix structure implemented by Nicolas Aunai. They are only provided here because they are required to compile and run the algorithm.

D.2.2 Parallel genetic algorithm

All files required for weights generation are located in the “GA” folder, in “Microarray”. In this folder, the files “genetic.c” and “main.c” are the only ones directly referring to the work implemented in this Thesis. Other files were developed by Christophe Duhamel, remain his property, and are only provided here because they are required to compile and run the genetic algorithm.
As an exemple, we also provide a set of weights, (“DATA/weights-MIT.txt”). With the makefile is provided, compile with “make release”.

D.3 Epigenetic model

All files required for the epigenetic model are available in the “Crypt model” folder.

Bibliography

Adjeroh, D. A., Zhang, Y., and Parthe, R. (2006). On denoising and compression of DNA microarray images. *Pattern Recognition*, 39(2006):2478–2493.

Anderson, R. M. (1988). The epidemiology of HIV infection: Variable incubation plus infectious periods and heterogeneity in sexual activity. *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, 151(1):66–98.

Baldazzi, V., Castiglione, F., and Bernaschi, M. (2006). An enhanced agent based model of the immune system response. *Cellular Immunology*, 244(2006):77–79.

Ball, F., Mollison, D., and Scalia-Tomba, G. (1997). Epidemics with two levels of mixing. *Annals of Applied Probability*, 87(1).

Barat, A., Ruskin, H. J., and Crane, M. (2006a). Probabilistic models for drug dissolution part 1: A review of Monte-Carlo & Cellular Automata approaches. *Simulation Modelling Practice and Theory*, 14(7):843–856.

Barat, A., Ruskin, H. J., and Crane, M. (2006b). Probabilistic models for drug dissolution part 2: Modelling a soluble binary drug delivery system dissolving in vitro. *Simulation Modelling Practice and Theory*, 14(7):857–873.

Barnes, M. G. and Weiss, A. A. (2003). Activation of the complement cascade by bordetella pertussis. *FEMS Microbiology Letters*, 220(2):271–275.

Barre-Sinoussi, F., Chermann, J. C., Rey, F., Nugeyre, M. T., Chamaret, S., Gruest, J.,

Dauguet, C., Axler-Blin, C., Vezinet-Brun, F., Rouzioux, C., Rozenbaum, W., and Montagnier, L. (1983). Isolation of a T-lymphotropic retrovirus from a patient at risk for acquired immune deficiency syndrome (AIDS). *Science*, 220(4599):868–871.

Barski, A., Cuddapah, S., Cui, K., Roh, T.-Y., Schones, D. E., Wang, Z., Wei, G., Chepelev, I., and Zhao, K. (2007). High-resolution proling of histone methylations in the human genome. *Cell*, 129:823–837.

Baylin, S. B. and Ohm, J. E. (2006). Epigenetic gene silencing in cancer – a mechanism for early oncogenic pathway addiction? *Nature Reviews Cancer*, 6:107–116.

Bednarik, D. P., Mosca, J. D., and Raj, N. B. (1987). Methylation as a modulator of expression of human immunodeficiency virus. *Journal of Virology*, 61(4):1253–1257.

Benenson, Y., Gil, B., Ben-Dor, U., Adar, R., and Shapiro, E. (2004). An autonomous molecular computer for logical control of gene expression. *Nature*, 429:423–429.

Benyoussef, A., HafidAllah, N. E., ElKenz, A., Ez-Zahraouy, H., and Loulidi, M. (2003). Dynamics of HIV infection on 2D cellular automata. *Physica A*, 322(2003):506–520.

Berlekamp, E. R., Conway, J. H., and Guy, R. K. (2004). *Winning Ways for your Mathematical Plays (2nd edition)*. Wellesley, Massachusetts: A. K. Peters Ltd.

Bernaschi, M. and Castiglione, F. (2001). Design and implementation of an immune system simulator. *Computers in Biology and Medicine*, 31(2001):303–331.

Biberfeld, P., Porwit-Ksiazek, A., Bottiger, B., Morfeldt-Mansson, L., and Biberfeld, G. (1985). Immunohistopathology of lymph nodes in HTLV-III infected homosexuals with persistent adenopathy or AIDS. *Cancer Research*, 45(9 suppl.):4665s–4670s.

Bird, A. (2002). DNA methylation patterns and epigenetic memory. *Genes & Development*, 16:6–21.

Bohringer, C. and Rutherford, T. F. (2008). Combining bottom-up and top-down. *Energy Economics*, 30(2):574–596.

Bond, A. H. and Gasser, L. (1988). *Readings in Distributed Artificial Intelligence*. Morgan Kaufman Publishers.

Brehm, M., Pinto, A., Daniels, K., Schneck, J., Welsh, R., and Selin, L. (2002). T cell immunodominance and maintenance of memory regulated by unexpectedly cross-reactive pathogens. *Nature Immunology*, 3:627–634.

Brenchley, J. M., Price, D. A., Schacker, T. W., Asher, T. E., Silvestri, G., Rao, S., Kazzaz, Z., Bornstein, E., Lambotte, O., Altmann, D., Blazar, B. R., Rodriguez, B., Teixeira-Johnson, L., Landay, A., Martin, J. N., Hecht, F. M., Picker, L. J., Lederman, M. M., Deeks, S. G., and Douek, D. C. (2006). Microbial translocation is a cause of systemic immune activation in chronic HIV infection. *Nature Medicine*, 12:1365–1371.

Brenchley, J. M., Schacker, T. W., Ruff, L. E., Price, D. A., Taylor, J. H., Beilman, G. J., Nguyen, P. L., Khoruts, A., Larson, M., Haase, A. T., and Douek, D. C. (2004). $CD4^+$ T cell depletion during all stages of HIV disease occurs predominantly in the gastrointestinal tract. *Journal of Experimental Medicine*, 200(6):749–759.

Brenner, C. (2005). *Le role des methyltransferases de l'ADN dans la regulation transcriptionnelle*. PhD thesis, Universite Libre de Bruxelles.

Burns, J. (2005). *Emergent networks in immune system shape space*. PhD thesis, Dublin City University, School of Computing.

Burns, J. and Ruskin, H. J. (2004). Network topology in immune system shape space. *Lecture Notes in Computer Science*, 3038:1094–1101.

Buseyne, F. and Riviere, Y. (2001). The flexibility of the TCR allows recognition of a large set of naturally occurring epitope variants by HIV-specific cytotoxic T lymphocytes. *International Immunology*, 13(7):941–950.

Busygin, S., Jacobsen, G., and Kramer, E. (2002). Double conjugated clustering applied to

leukemia microarray data. In *Proceedings of the 2nd SIAM International Conference on Data Mining, Workshop on Clustering High Dimensional Data*.

Cahill, R. N. P., Frost, H., and Trnka, Z. (1976). The effects of antigen on the migration of recirculating lymphocytes through single lymph node. *Journal of Experimental Medicine*, 143:870–888.

Cammarata, S., McArthur, D., and Steeb, R. (1983). Strategies of cooperation in distributed problem solving. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence (IJCAI-83)*, Karlsruhe, Germany.

Cantu-Paz, E. (1995). A summary of research on parallel genetic algorithms. *IlliGAL report 95007*, University of Illinois (IL).

Carneiro, J. and Stewart, J. J. (1994). Rethinking Shape Space: Evidence from simulated docking suggests that steric complementarity is not limiting for antibody-antigen recognition and idiotypic interactions. *Journal of Theoretical Biology*, 169:391–402.

Castiglione, F., Poccia, F., D'Offizi, G., and Bernaschi, M. (2004). Mutation, fitness, viral diversity, and predictive markers of disease progression in a computational model of HIV type 1 infection. *AIDS Research and Human Retroviruses*, 20(12):1314–1323.

Cedeno, W. and Vemuri, V. (1993). An investigation of DNA mapping with genetic algorithms: preliminary results. In *Proceedings of the Fifth Workshop on Neural Networks*, Vol. 2204 of SPIE.

Celada, F. and Seiden, P. E. (1992). A computer model of cellular interactions in the immune system. *Immunology Today*, 13(2):56–62.

Chaib-Draa, B. and Dignum, F. (2002). Trends in agent communication language. *Computational Intelligence*, 5(2).

Chaib-Draa, B., Jarras, I., and Moulin, B. (2001). Systemes multi-agents : principes gener-

aux et applications. In Briot, J. and Demazeau, Y., editors, *Principes et architectures des systemes multi-agents*, chapter 1, pages 27–70. Hermes Science.

Chakraborty, A. and Maka, H. (2005). Biclustering of gene expression data using genetic algorithm. In *Proceedings of the 2005 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*.

Chan, D. C., Fass, D., Berger, J. M., and Kim, P. S. (1997). Core structure of gp41 from the HIV envelope glycoprotein. *Cell*, 89(2):263–273.

Chang, S. and Aune, T. M. (2007). Dynamic changes in histone-methylation 'marks' across the locus encoding interferon-γ during the differentiation of t helper type 2 cells. *Nature Immunology*, 8:723–731.

Chen, C., Xia, J., Liu, J., and Feng, G. (2006). Nonlinear inversion of potential-field data using a hybrid-encoding genetic algorithm. *Computers & Geosciences*, 32(2):230–239.

Cheng, Y. and Church, G. M. (2000). Biclustering of expression data. In *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology (ISMB 2000)*, volume 8, San Diego, California, USA.

Cheynier, R., Henrichwark, S., Hadida, F., Pelletier, E., Oksenhendler, E., Autran, B., and Wain-Hobson, S. (1994). HIV and T cell expansion in splenic white pulps is accompanied by infiltration of HIV-specific cytotoxic T lymphocytes. *Cell*, 78(3):373–387.

Cimarelli, A. and Darlix, J.-L. (2002). Assembling the human immunodeficiency virus type 1. *Cellular and Molecular Life Sciences*, 59(7):1166–1184.

Cloonan, N., Forrest, A. R. R., Kolle, G., Gardiner, B. B. A., Faulkner, G. J., Brown, M. K., Taylor, D. F., Steptoe, A. L., Wani, S., Bethel, G., Robertson, A. J., Perkins, A. C., Bruce, S. J., Lee, C. C., Ranade, S. S., Peckham, H. E., Manning, J. M., McKernan, K. J., and Grimmond, S. M. (2008). Stem cell transcriptome profiling via massive-scale mRNA sequencing. *Nature Methods*, 5(7):613–619.

Cooney, C. A., Dave, A. A., and Wolff, G. L. (2002). Maternal methyl supplements in mice affect epigenetic variation and DNA methylation of offspring. *Journal of Nutrition*, 132:2393S–2400S.

Crick, F. (1970). Central dogma of molecular biology. *Nature*, 227:561–563.

Crutchfield, J. P. (1994). The calculi of emergence: computation, dynamics and induction. *Physica D: Nonlinear Phenomena*, 75(1-3):11–54.

Das, R., Dimitrova, N., Xuan, Z., Rollins, R. A., Haghighi, F., Edwards, J. R., Ju, J., Bestor, T., and Zhang, M. Q. (2006). Computational prediction of methylation status in human genomic sequences. *Proceedings of the National Academy of Sciences USA*, 103(28):10713–10716.

Davey, C., Fraser, R., Smolle, M., Simmen, M. W., and Allan, J. (2003). Nucleosome positioning signals in the DNA sequence of the human and mouse H19 imprinting control regions. *Journal of Molecular Biology*, 325(5):873–887.

de Saint-Vis, B., Cupillard, L., Pandrau-Garcia, D., Ho, S., Renard, N., Grouard, G., Duvert, V., Thomas, X., Galizzi, J. P., and Banchereau, J. (1995). Distribution of carboxypeptidase M on lymphoid and myeloid cells parallels the other zinc-dependent proteases CD10 and CD13. *Blood*, 86(3):1098–1105.

Delelis, O., Lehmann-Che, J., and Saib, A. (2004). Foamy viruses: a world apart. *Current Opinion in Microbiology*, 7:400–406.

Derdeyn, C. A. and Silvestri, G. (2005). Viral and host factors in the pathogenesis of HIV infection. *Current Opinion in Immunology*, 17(4):366–373.

Diestel, R. (2005). *Graph Theory (Graduate Texts in Mathematics, Volume 173)*. Springer.

Dodge, J. E., Ramsahoye, B. H., Wo, Z. G., Okano, M., and Li, E. (2002). De novo methylation of MMLV provirus in embryonic stem cells: CpG versus non-CpG methylation. *Gene*, 289(1–2):41–48.

Douek, D. C., McFarland, R. D., H.Keiser, P., Gage, E. A., Massey, J. M., Haynes, B. F., Polis, M. A., Haasek, A. T., Feinberg, M. B., Sullivan, J. L., Jamieson, B. D., Zack, J. A., Picker, L. J., and Koup, R. A. (1998). Changes in thymic function with age and during the treatment of HIV infection. *Nature*, 396:690–695.

Douek, D. C., Picker, L. J., and Koup, R. A. (2003). T cell dynamics in HIV-1 infection. *Annual Review of Immunology*, 21:265–304.

Dove, A. (2006). Virtual models best live cells at predicting biology. *Nature Medecine*, 12:1225.

Durfee, E. H. (1998). *Coordination of distributed problem solvers*. Kluwer Academic Publishers.

Durfee, E. H. (2001). Scaling up agent coordination strategies. *Computer*, 34(7):39–46.

Farooqi, Z. H. and Mohler, R. R. (1989). Distribution models of recirculating lymphocytes. *IEEE Transactions on Biomedical Engineering*, 36(3):355–362.

Fatemi, M., Pao, M. M., Jeong, S., Gal-Yam, E. N., Egger, G., Weisenberger, D. J., and Jones, P. A. (2005). Footprinting of mammalian promoters: use of a CpG DNA methyltransferase revealing nucleosome positions at a single molecule level. *Nucleic Acids Research*, 33(20):e176.

Fickett, J. and Cinkosky, M. (1993). A genetic algorithm for assembling chromosome physical maps. In *Proceedings of the Second International Conference on Bioinformatics, Supercomputing, and Complex Genome Analysis*.

Forrest, S. and Hofmeyr, S. A. (2001). Immunology as information processing. In Segel, L. A. and Cohen, I., editors, *Design Principles for the Immune System and Other Distributed Autonomous Systems*, chapter 12, pages 361–387. Oxford University Press.

Franklin, S. and Graesser, A. (1997). Is it an agent, or just a program? : A taxonomy for autonomous agents. In Mueller, J., Wooldridge, M., and Jennings, N., editors, *Intelligent*

Agents III : Theories, Architectures, and Languages (LNAI Volume 1193), pages 21–35. Springer-Verlag.

Fujikado, N., Saijo, S., and Iwakura, Y. (2006). Identification of arthritis-related gene clusters by microarray analysis of two independent mouse models for rheumatoid arthritis. *Arthritis Research & Therapy*, 8(4):R100.

Garber, D. A., Silvestri, G., and Feinberg, M. B. (2004). Prospects for an AIDS vaccine: three big questions, no easy answers. *The Lancet Infectious Diseases*, 4(7):397–413.

Gardiner-Garden, M. and Frommer, M. (1987). CpG islands in vertebrate genomes. *Journal of Molecular Biology*, 196(2):261–282.

Gardner, M. (1970). Mathematical games: The fantastic combinations of John Conway's new solitaire game Life. *Scientific American*, 223:120–123.

Garvy, B. A. (2003). Host defense against pulmonary infection in neonates. *Clinical and Applied Immunology Reviews*, 4(2003):205–223.

Gaudet, F., Hodgson, J. G., Eden, A., Jackson-Grusby, L., Dausman, J., Gray, J. W., Leonhardt, H., and Jaenisch, R. (2003). Induction of tumors in mice by genomic hypomethylation. *Science*, 300:489–492.

Germain, R. N. (2001). The art of the probable: System control in the adaptive immune system. *Science*, 293(5528):240–245.

Gilmore, J. M. and Washburn, M. P. (2007). Deciphering the combinatorial histone code. *Nature Methods*, 4(6):480–481.

Goertzel, B. (1992). Self-organizing evolution. *Journal of Social and Evolutionary Systems*, 15(1):7–53.

Goldberg, D. E. and Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. In Rawlins, G., editor, *Foundations of Genetic Algorithms*, chapter 12, pages 69–93. Morgan Kaufmann Publishers.

Gray, H. (1918). *Anatomy of the Human Body (20th U.S. edition)*. Philadelphia: Lea & Febiger.

Greenberg, S. A. (2001). DNA microarray gene expression analysis technology and its application to neurological disorders. *Neurology*, 57:755–761.

Grefenstette, J. (1981). Parallel adaptive algorithms for function optimization. *Technical report CS-81-19*, Vanderbilt University (TN).

Gropp, W., Lusk, E., Doss, N., and Skjellum, A. (1996). A high-performance, portable implementation of the MPI message passing interface standard. *Parallel Computing*, 22(6):789–828.

Gropp, W., Lusk, E., and Skjellum, A. (1999a). *Using MPI-2: Advanced Features of the Message Passing Interface*. MIT Press.

Gropp, W., Lusk, E., and Skjellum, A. (1999b). *Using MPI: Portable Parallel Programming With the Message-Passing Interface, second edition*. MIT Press.

Grossman, Z., Feinberg, M. B., and Paul, W. E. (1998). Multiple modes of cellular activation and virus transmission in HIV infection: a role for chronically and latently infected cells in sustaining viral replication. *Proceedings of the National Academy of Sciences USA*, 95(11):6314–6319.

Guadalupe, M., Reay, E., Sankaran, S., Prindiville, T., Flamm, J., McNeil, A., and Dandekar, S. (2003). Severe $CD4^+$ T-cell depletion in gut lymphoid tissue during primary human immunodeciency virus type 1 infection and substantial delay in restoration following highly active antiretroviral therapy. *Journal of Virology*, 77(21):11708–11717.

Harper, D. (2001). *Online etymology dictionary*. `http://www.etymonline.com/index.php`.

Hayes-Roth, B., Hewett, M., Washington, R., Hewett, R., and Seiver, A. (1989). Distributing intelligence within an individual. In Gasser, L. and Huhns, M., editors, *Distributed*

Artificial Intelligence Volume II, pages 385–412. Pitman Publishing and Morgan Kaufmann.

Hazenberg, M. D., Hamann, D., Schuitemaker, H., and Miedema, F. (2000). T cell depletion in HIV-1 infection: how CD4+ T cells go out of stock. *Nature Immunology*, 1:285–289.

Hecquet, D., Ruskin, H. J., and Crane, M. (2007). Optimisation and parallelisation strategies for Monte Carlo simulation of HIV infection. *Computers in Biology and Medicine*, 37(5):691–699.

Helsinger, A., Kleinmann, K., and Brinn, M. (2004). A framework to control emergent survivability of multi agent systems. In *Proceedings of Third International Joint Conference on Autonomous Agents and Multiagent Systems*, volume 1, pages 28–35.

Hill, T., Lundgren, A., Fredriksson, R., and Schioth, H. B. (2005). Genetic algorithm for large-scale maximum parsimony phylogenetic analysis of proteins. *Biochimica Biophysica Acta*, 1725(1):19–29.

Holland, J. H. (1975). *Adaptation in natural and artificial systems*. MIT Press.

Horwitz, P. and Wilcox, B. A. (2005). Parasites, ecosystems and sustainability: an ecological and complex systems perspective. *International Journal for Parasitology*, 35(7):725–732.

Howard, M. and Paul, W. E. (1982). Interleukins for B lymphocytes. *Lymphokine Research*, 1(1):1–4.

Iglesias, C. A., Garijo, M., Gonzalez, J. C., , and Velasco, J. R. (1997). Analysis and design of multiagent systems using mas-commonkads. In *AAAI'97 Workshop on Agent Theories, Architectures and Languages*, Providence, RI, USA.

Ishida, T., Hamano, A., Koiwa, T., and Watanabe, T. (2006). 5' long terminal repeat (LTR)-selective methylation of latently infected HIV-1 provirus that is demethylated by reactivation signals. *Retrovirology*, 3:69.

Issa, J.-P. J., Ottaviano, Y. L., Celano, P., Hamilton, S. R., Davidson, N. E., and Baylin, S. B. (1994). Methylation of the oestrogen receptor CpG island links ageing and neoplasia in human colon. *Nature Genetics*, 7:536–540.

Jackson, W. C. and Norgard, J. D. (2008). A hybrid genetic algorithm with Boltzmann convergence properties. *Journal of Optimization Theory and Applications*, 136(3):431–443.

Janssen, M. A. and Ostrom, E. (2006). Governing social-ecological systems. In Tesfatsion, L. and Judd, K., editors, *Handbook of Computational Economics*, chapter 30, pages 1465–1509.

Jennings, N., Sycara, K., and Wooldridge, M. (1998). A roadmap of agent research and development. *Autonomous agents and multi-agents systems*, 1(1):7–38.

Jensen, K., Talbot, R., Paxton, E., Waddington, D., and Glass, E. J. (2006). Development and validation of a bovine macrophage specific cDNA microarray. *BMC Genomics*, 7:224.

Jenuwein, T. and Allis, C. D. (2001). Translating the histone code. *Science*, 293(5532):1074–1080.

Jones, P. A. (2002). DNA methylation and cancer. *Oncogene*, 21:5358–5360.

Kaneda, A., Kaminishi, M., Yanagihara, K., Sugimura, T., and Ushijima, T. (2002). Identication of silencing of nine genes in human gastric cancers. *Cancer Research*, 62:6645–6650.

Kasahara, M., Suzuki, T., and du Pasquier, L. (2004). On the origins of the adaptive immune system: novel insights from invertebrates and cold-blooded vertebrates. *Trends in Immunology*, 25(2):105–111.

Katayama, K., Hirabayashi, H., and Narihisa, H. (2003). Analysis of crossovers and se-

lections in a coarse-grained parallel genetic algorithm. *Mathematical and Computer Modelling*, 38(11-13):1275–1282.

Kaufmann, S. H. E. (1999). Cell-mediated immunity: Dealing a direct blow to pathogens. *Current Biology*, 9(3):R97–R99.

Kerr, G., Ruskin, H. J., and Crane, M. (2008). Pattern discovery in gene expression data. In Wang, H.-F., editor, *Intelligent Data Analysis: Developing New Methodologies Through Pattern Discovery and Recovery*, chapter 3. Information Science Reference.

Kinny, D. and Georgeff, M. (1996). Modelling and design of multi-agents systems. In *Intelligent agents III: Proceedings of the third international workshop on agent theories, architectures and languages*, Budapest, Hungary.

Koch, C. M., Andrews, R. M., Flicek, P., Dillon, S. C., Karaoz, U., Clelland, G. K., Wilcox, S., Beare, D. M., Fowler, J. C., Couttet, P., James, K. D., Lefebvre, G. C., Bruce, A. W., Dovey, O. M., Ellis, P. D., Dhami, P., Langford, C. F., Weng, Z., Birney, E., Carter, N. P., Vetrie, D., and Dunham, I. (2007). The landscape of histone modifications across 1% of the human genome in five human cell lines. *Genome Research*, 17(6):691–707.

Kondo, Y., Kanai, Y., Sakamoto, M., Mizokami, M., Ueda, R., and Hirohashi, S. (2000). Genetic instability and aberrant DNA methylation in chronic hepatitis and cirrhosis – A comprehensive study of loss of heterozygosity and microsatellite instability at 39 loci and DNA hypermethylation on 8 CpG islands in microdissected specimens from patients with hepatocellular carcinoma. *Hepatology*, 32:970–979.

Kubota, T. (2008). Epigenetics in congenital diseases and pervasive developmental disorders. *Environmental Health and Preventive Medicine*, 13(1):3–7.

Kunkel, E. J. and Butcher, E. C. (2002). Chemokines and the tissue-specific migration of lymphocytes. *Immunity*, 16:1–4.

Langerman, J. J. and Ehlers, E. M. (1997). Agent-based airline scheduling. *Computers & Industrial Engineering*, 33(3–4):849–852.

Lever, A. M. L. (2005). HIV: the virus. *Medicine*, 33(6):1–3.

Levine, D. (1994). A parallel genetic algorithm for the set partitioning problem. *Technical report ANL-94/23*, University of Illinois (IL).

Levy, J. A. (1990). The importance of the innate immune system in controlling HIV infection and disease. *Trends in Immunology*, 22(6):312–316.

Liou, J., Wu, M., Lin, J., Wang, H., Huang, S., Chiu, H., Lee, Y., Lin, Y., Shun, C., and Liang, J. (2007). Loss of imprinting of insulin-like growth factor II is associated with increased risk of proximal colon cancer. *European Journal of Cancer*, 43(8):1276–1282.

Little, S. J., Holte, S., Routy, J.-P., Daar, E. S., Markowitz, M., Collier, A. C., Koup, R. A., Mellors, J. W., Connick, E., Conway, B., Kilby, M., Wang, L., Whitcomb, J. M., Hellmann, N. S., and Richman, D. D. (2002). Antiretroviral-drug resistance among patients recently infected with HIV. *The New England Journal of Medicine*, 347(6):385–394.

Liu, J. and Wang, W. (2003). OP-cluster: clustering by tendency in high dimensional space. In *Proceedings of the 3rd IEEE International Conference on Data Mining*, pages 187–194.

MacDonell, K. B., Chmiel, J. S., Poggensee, L., Wu, S., and Phair, J. P. (1990). Predicting progression to AIDS: Combined usefulness of CD4 lymphocyte counts and p24 antigenemia. *The American Journal of Medicine*, 89(6):706–712.

Mackay, C. R., Marston, W. L., and Dudler, L. (1990). Naive and memory T cells show distinct pathways of lymphocyte recirculation. *Journal of Experimental Medicine*, 171:801–817.

Madeira, S. C. and Oliveira, A. L. (2004). Biclustering algorithms for biological data anal-

ysis: a survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1(1):24–45.

Maekita, T., Nakazawa, K., Mihara, M., Nakajima, T., Yanaoka, K., Iguchi, M., Arii, K., Kaneda, A., Tsukamoto, T., Tatematsu, M., Tamura, G., Saito, D., Sugimura, T., Ichinose, M., and Ushijima, T. (2006). High levels of aberrant DNA methylation in Helicobacter pylori-infected gastric mucosae and its possible association with gastric cancer risk. *Clinical Cancer Research*, 12:989–995.

Mamidala, A. R., Chai, L., Jin, H.-W., and Panda, D. K. (2006). Efficient SMP-aware MPI-level broadcast over InfiniBands hardware multicast. In *Workshop on Communication Architecture for Clusters, 20th International Parallel and Distributed Processing Symposium (IPDPS 2006)*.

Mannion, R., Ruskin, H. J., and Pandey, R. B. (2000). Effect of mutation on helper T-cells and viral population: a computer simulation model for HIV. *Theory Biosciences*, 119(2000):145–155.

Mannion, R., Ruskin, H. J., and Pandey, R. B. (2002). A Monte-Carlo approach to population dynamics of cells in a HIV immune response model. *Theory in Biosciences*, 121(2002):237–245.

Mano, H. (2008). Epigenetic abnormalities in cardiac hypertrophy and heart failure. *Environmental Health and Preventive Medicine*, 13(1):25–29.

Martin, M. P. and Carrington, M. (2005). Immunogenetics of viral infections. *Current Opinion in Immunology*, 17(5):510–526.

Mathe, G., Colasante, U., Morette, C., Hallard, M., and Blanquet, D. (1996). Will killing the last HIV-1 particle cure AIDS patients? II: second part. decrease of viral load and of T-suppressor cells, and increase of the cytotoxic cells, without effect on CD4, after the use of 10 virostatics applied in 3 or 4 drug combinations of different sequences. the time for CD4 immunotherapy? *Biomedecine & Pharmacotherapy*, 50(10):473–479.

Matsumoto, M. and Nishimura, T. (1998). Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation*, 8(1):3–30.

Mattapallil, J. J., Douek, D. C., Hill, B., Nishimura, Y., Martin, M., and Roederer, M. (2005). Massive infection and loss of memory $CD4^+$ T cells in multiple tissues during acute SIV infection. *Nature*, 434:1093–1097.

McCarthy, I. P. and Tan, Y. K. (2000). Manufacturing competitiveness and fitness landscape theory. *Journal of Materials Processing Technology*, 107(1-3):347–352.

McCune, J. M. (1997). Thymic function in HIV-1 disease. *Seminars in Immunology*, 9(6):397–404.

McGowan, P. O. and Kato, T. (2008). Epigenetics in mood disorders. *Environmental Health and Preventive Medicine*, 13(1):16–24.

Mehandru, S., Poles, M. A., Tenner-Racz, K., Horowitz, A., Hurley, A., Hogan, C., Boden, D., Racz, P., and Markowitz, M. (2005a). Primary HIV-1 infection is associated with preferential depletion of $CD4^+$ T lymphocytes from effector sites in the gastrointestinal tract. *Journal of Experimental Medicine*, 200(6):761–770.

Mehandru, S., Tenner-Racz, K., Racz, P., and Markowitz, M. (2005b). The gastrointestinal tract is critical to the pathogenesis of acute HIV-1 infection. *Journal of Allergy and Clinical Immunology*, 116:419–422.

Minar, N., Burkhart, R., Langton, C., and Askenazi, M. (1996). The Swarm simulation system: A toolkit for building multi-agent simulations. *Working Paper 96-06-042, Santa Fe Institute*.

Mitra, S. and Banka, H. (2006). Multi-objective evolutionary biclustering of gene expression data. *Pattern Recognition*, 39(2006):2464–2477.

Mizoguchi, F., Nishiyama, H., Ohwada, H., and Hiraishi, H. (1999). Smart office robot collaboration based on multi-agent programming. *Artificial Intelligence*, 114(1999):57–94.

Mortazavi, A., Williams, B. A., McCue, K., Schaeffer, L., and Wold, B. (2008). Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nature Methods*, 5(7):621–628.

Mucha, L., Stephenson, J., Morandi, N., and Dirani, R. (2006). Meta-analysis of disease risk associated with smoking, by gender and intensity of smoking. *Gender Medicine*, 3(4):279–291.

Munier, M. L. and Kelleher, A. D. (2007). Acutely dysregulated, chronically disabled by the enemy within: T-cell responses to HIV-1 infection. *Immunology and Cell Biology*, 85:6–15.

Murali-Krishna, K., Lau, L. L., Sambhara, S., Lemonnier, F., Altman, J., and Ahmed, R. (1999). Persistence of memory CD8 T cells in MHC Class I-deficient mice. *Science*, 286(5443):1377–1381.

Murphy, K. M., Travers, P., and Walport, M. (2007). Evolution of the immune system. In *Janeways Immunobiology (seventh edition)*, chapter 16. Garland Science Publishing.

Nabel, G. and Baltimore, D. (1987). An inducible transcription factor activates expression of human immunodeficiency virus in T cells. *Nature*, 326(6114):711–713.

Nakajima, T., Enomoto, S., and Ushijima, T. (2008). DNA methylation: a marker for carcinogen exposure and cancer risk. *Environmental Health and Preventive Medicine*, 13(1):8–15.

Nakajima, T., Maekita, T., Oda, I., Gotoda, T., Yamamoto, S., Umemura, S., Ichinose, M., Sugimura, T., Ushijima, T., and Saito, D. (2006a). Higher methylation levels in gastric

mucosae significantly correlate with higher risk of gastric cancers. *Cancer Epidemiology Biomarkers & Prevention*, 15:2317–2321.

Nakajima, T., Oda, I., Gotoda, T., Hamanaka, H., Eguchi, T., Yokoi, C., and Saito, D. (2006b). Metachronous gastric cancers after endoscopic resection: how effective is annual endoscopic surveillance? *Gastric Cancer*, 9(2):93–98.

Naresh, R., Tripathi, A., and Omar, S. (2006). Modelling the spread of AIDS epidemic with vertical transmission. *Applied Mathematics and Computation*, 178(2):262–272.

Ndifon, W. (2005). A complex adaptive systems approach to the kinetic folding of RNA. *Biosystems*, 82(3):257–265.

Nimura, K., Ishida, C., Koriyama, H., Hata, K., Yamanaka, S., Li, E., Ura, K., and Kaneda, Y. (2001). Dnmt3a2 targets endogenous Dnmt3L to ES cell chromatin and induces regional DNA methylation. *Genes to Cells*, 11:1225–1237.

Ohagen, A., Luftig, R. B., Reicin, A. S., Yin, L., Ikuta, K., Kimura, T., Goff, S. P., and Hoglund, S. (1997). The morphology of the immature HIV-1 virion. *Virology*, 228(1):112–114.

Oxenius, A., Price, D., Dawson, S. J., Tun, T., Easterbrook, P. J., Phillips, R. E., and Sewell, A. K. (2001). Cross-staining of cytotoxic T lymphocyte populations with peptide-MHC class I multimers of natural HIV-1 variant antigens. *AIDS*, 15(1):121–122.

Pandey, R. B., Mannion, R., and Ruskin, H. J. (2000). Effect of cellular mobility on immune response. *Physica A*, 283(2000):447–450.

Papa, A. R. and Tsallis, C. (1996). A local-field-type model for immunological systems: time evolution in real and shape space. *Physica A*, 233(1-2):85–101.

Parsons, R. J., Forrest, S., and Burks, C. (1995). Genetic algorithms, operators, and DNA fragment assembly. *Machine Learning*, 21:11–33.

Pathak, S. and Palan, U. (2005). *Immunology: Essential and Fundamental*. Science Publishers.

Pathscale^{TM}(2005). *PathScale InfiniPath^{TM}HTX^{TM}Adapter*. Technical documentation.

Peeters, R. (2003). The maximum edge biclique problem is NP-complete. *Discrete Applied Mathematics*, 131(3):651–654.

Pennings, S., Allan, J., and Davey, C. S. (2005). DNA methylation, nucleosome formation and positioning. *Briefings in Functional Genomics and Proteomics*, 3(4):351–361.

Pereira, C. M. N. A. and Lapa, C. M. F. (2003). Coarse-grained parallel genetic algorithmnext term applied to a nuclear reactor core design optimization problem. *Annals of Nuclear Energy*, 30(5):555–565.

Perelson, A. S. and Nelson, P. W. (1999). Mathematical analysis of HIV-1 dynamics in vivo. *SIAM Review*, 41(1):3–44.

Perelson, A. S. and Oster, G. F. (1979). Theoretical studies of clonal selection: Minimal antibody repertoire size and reliability of self-non-self discrimination. *Journal of Theoretical Biology*, 81(4):645–670.

Perrin, D. and Burns, J. (2008). Large-scale immune models and visualization. *ERCIM News*, 74:33–34.

Perrin, D., Burns, J., Ruskin, H. J., and Crane, M. (2006a). An agent-based approach to immune modelling. *Lecture Notes in Computer Science*, 3980:612–621.

Perrin, D., Ruskin, H. J., and Crane, M. (2006b). An agent-based approach to immune modelling: Priming individual response. In *Selected proceedings of Third International Conference on Modeling and Simulation (ICMS06)*, Cairo, Egypt.

Perrin, D., Ruskin, H. J., and Crane, M. (2006c). HIV modelling - a parallel implementation of a lymph network. In *Selected proceedings of Third International Conference on Cluster and Grid Computing Systems (CGCS 2006)*, Venice, Italy.

Perrin, D., Ruskin, H. J., Crane, M., and Walshe, R. (2008). Epigenetic modelling. *ERCIM News*, 72:46.

Pohlmann, S., Baribaud, F., and Doms, R. W. (2001). DC-SIGN and DC-SIGNR: helping hands for HIV. *Trends in Immunology*, 22(12):643–464.

Potmesil, M. and Goldfeder, A. (1973). Nucleolar morphology and cell proliferation kinetics of thymic lymphocytes. *Experimental Cell Research*, 77(1-2):31–40.

Potvin, J.-Y., Duhamel, C., and Guertin, F. (1996). A genetic algorithm for vehicle routing with backhauling. *Computational Statistics and Data Analysis*, 6(4):345–355.

Press, W. H., Vetterling, W. T., Teukolsky, S. A., and Flannery, B. P. (2002). *Numerical Recipes in C++: the art of Scientific Computing*. Cambridge University Press.

Prusty, B. K. and Das, B. C. (2005). Constitutive activation of transcription factor AP-1 in cervical cancer and suppression of human papillomavirus (HPV) transcription and AP-1 activity in HeLa cells by curcumin. *International Journal of Cancer*, 113(6):951–960.

Rainier, S., Johnson, L. A., Dobry, C. J., Ping, A. J., Grundy, P. E., and Feinberg, A. P. (1993). Relaxation of imprinted genes in human cancer. *Nature*, 362:747–749.

Raychaudhuri, S., Sutphin, P. D., Chang, J. T., and Altman, R. B. (2001). Basic microarray analysis: grouping and feature reduction. *Trends in Biotechnology*, 19:189–193.

Reeves, J. D. and Doms, R. W. (2002). Human immunodeficiency virus type 2. *Journal of General Virology*, 83:1253–1265.

Rho, H. M., Poiesz, B., Ruscetti, F. W., and Gallo, R. C. (1981). Characterization of the reverse transcriptase from a new retrovirus (HTLV) produced by a human cutaneous T-cell lymphoma cell line. *Virology*, 112(1):355–360.

Robertson, D. L., Hahn, B. H., and Sharp, P. M. (1995). Recombination in AIDS viruses. *Journal of Molecular Evolution*, 40(3):249–259.

Roederer, M., Dubs, J. G., Anderson, M. T., Raju, P. A., Herzenberg, L. A., and Herzenberg, L. A. (1995). CD8 naive T cell counts decrease progressively in HIV-infected adults. *Journal of Clinical Investigation*, 95(5):2061–2066.

Rosok, B. I., Bostad, L., Voltersvik, P., Bjerknes, R., Olofsson, J., Asjo, B., and Brinchmann, J. E. (1996). Reduced CD4 cell counts in blood do not reflect CD4 cell depletion in tonsillar tissue in asymptomatic HIV-1 infection. *AIDS*, 10(10):F35–F38.

Ruskin, H. J. and Burns, J. (2005). Network emergence in immune system shape space. *Lecture Notes in Computer Science*, 3481:1254–1263.

Ruskin, H. J. and Burns, J. (2006). Weighted networks in immune system shape space. *Physica A*, 365(2):549–555.

Ruskin, H. J., Pandey, R. B., and Liu, Y. (2002). Viral load and stochastic mutation in a Monte Carlo simulation of HIV. *Physica A*, 311(1–2):213–220.

Sacks, S. H., Chowdhury, P., and Zhou, W. (2003). Role of the complement term system in rejection. *Current Opinion in Immunology*, 15(5):487–492.

Sanders, V. M. (2006). Epigenetic regulation of th1 and th2 cell development. *Brain, Behavior, and Immunity*, 20(4):317–324.

Sarson, A. J., Read, L. R., Haghighi, H. R., Lambourne, M. D., Brisbin, J. T., Zhou, H., and Sharif, S. (2007). Construction of a microarray specific to the chicken immune system: profiling gene expression in B cells after lipopolysaccharide stimulation. *Canadian Journal of Veterinary Research*, 71(2):108–118.

Sattentau, Q. (2008). HIV's gut feeling. *Nature Immunology*, 9(3):225–227.

Saxonov, S., Berg, P., and Brutlag, D. L. (2006). A genome-wide analysis of CpG dinucleotides in the human genome distinguishes two distinct classes of promoters. *Proceedings of the National Academy of Sciences USA*, 103(5):1412–1417.

Schieferdecker, H. L., Ullrich, R., Hiserland, H., and Zeitz, M. (1992). T cell differentiation antigens on lymphocytes in the human intestinal lamina propria. *Journal of Immunology*, 149(8):2816–2822.

Schulze-Forster, K., Gotz, F., Wagner, H., Kroger, H., and Simon, D. (1990). Transcription of HIV1 is inhibited by DNA methylation. *Biochemical and Biophysical Research Communications*, 168(1):141–147.

Schumacher, A., Kapranov, P., Kaminsky, Z., Flanagan, J., Assadzadeh, A., Yau, P., Virtanen, C., Winegarden, N., Cheng, J., Gingeras, T., and Petronis, A. (2006). Microarray-based DNA methylation profiling: technology and applications. *Nucleic Acids Research*, 34(2):528–542.

Segal, E., Fondufe-Mittendorf, Y., Chen, L., Thastrom, A. C., Field, Y., Moore, I. K., Wang, J.-P. Z., and Widom, J. (2006). A genomic code for nucleosome positioning. *Nature*, 442:772–778.

Sei, S. (2005). Peptide nucleic acids as epigenetic inhibitors of HIV-1. *International Journal of Peptide Research and Therapeutics*, 10(3):269–286.

Seiden, P. and Celada, F. (1992). A model for simulating cognate recognition and response in the immune system. *Journal of theoretical Biology*, 158:329–357.

Sharkasi, A., Crane, M., Ruskin, H. J., and Matos, J. A. O. (2006). The reaction of stock markets to crashes and events: A comparison study between emerging and mature markets using wavelet transforms. *Physica A*, 368(2):511–521.

Shen, Q., Uray, I. P., Li, Y., Krisko, T., Strecker, T. E., Kim, H.-T., and Brown, P. H. (2008). The AP-1 transcription factor regulates breast cancer cell growth via cyclins and E2F factors. *Oncogene*, 27:366–377.

Shendure, J. (2008). The beginning of the end for microarrays? *Nature Methods*, 5(7):585–587.

Sierra, S., Kupfer, B., and Kaiser, R. (2005). Basics of the virology of HIV-1 and its replication. *Journal of Clinical Virology*, 34(2005):233–244.

Slonim, D. K. (2002). From patterns to pathways: gene expression data analysis comes of age. *Nature Genetics*, 32:502–508.

Smet, C. D., Lurquin, C., Lethe, B., Martelange, V., and Boon, T. (1999). DNA methylation is the primary silencing mechanism for a set of germ line- and tumor-specic genes with a CpG-rich promoter. *Molecular and Cellular Biology*, 19:7327–7335.

Somasundaram, K., Mungamuri, S. K., and Wajapeyee, N. (2002). DNA microarray technology and its applications in cancer biology. *Applied Genomics and Proteomics*, 1:209–218.

Speybroeck, L. V. (2000). The organism: A crucial genomic context in molecular epigenetics? *Theory in Biosciences*, 119(2000):187–208.

Spurny, K. R. (1996). Chemical mixtures in atmospheric aerosols and their correlation to lung diseases and lung cancer occurence in the general population. *Toxicology Letters*, 88:271–277.

Srikusalanukul, W., Bruyne, F. D., and McCullagh, P. (2000). Modelling of peripheral lymphocyte migration: System identification approach. *Immunology and Cell Biology*, 78(3):288–293.

Srinivasan, A., Mascagni, M., and Ceperley, D. (2003). Testing parallel random number generators. *Parallel Computing*, 29(2003):69–94.

Stekel, D. J., Parker, C. E., and Novak, M. A. (1997). A model of lymphocyte recirculation. *Immunology Today*, 18(5):216–221.

Sterne, J. A. C., Hernan, M. A., Ledergerber, B., Tilling, K., Weber, R., Sendi, P., Rickenbach, M., Robins, J. M., Egger, M., and the Swiss HIV Cohort Study (2005). Long-term

effectiveness of potent antiretroviral therapy in preventing AIDS and death: a prospective cohort study. *The Lancet*, 366(9483):378–384.

Stolovitzky, G. (2003). Gene selection in microarray data: the elephant, the blind men and our algorithms. *Current Opinion in Structural Biology*, 13:370–376.

Strahl, B. D. and Allis, C. D. (2000). The language of covalent histone modifications. *Nature*, 403:41–45.

Sycara, K. P. (1998). Multiagent systems. *AI magazine*, 19(2):79–92.

Tanay, A., Sharan, R., and Shamir, R. (2002). Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18(Suppl. 1):S136–S144.

Tesselaar, K., Arens, R., van Schijndel, G. M. W., Baars, P. A., van der Valk, M. A., Borst, J., van Oers, M. H. J., and van Lier, R. A. W. (2002). Lethal T cell immunodeficiency induced by chronic costimulation via CD27-CD70 interactions. *Nature Immunology*, 4:49–54.

Thurner, S. and Biely, C. (2007). The eigenvalue spectrum of lagged correlation matrices. *Acta Physica Polonica*, 38(13):4111–4122.

Tiguert, R., Gheiler, E. L., Tefilli, M. V., Oskanian, P., Banerjee, M., Grignon, D. J., Sakr, W., Pontes, J. E., and Jr, D. P. W. (1999). Lymph node size does not correlate with the presence of prostate cancer metastasis. *Urology*, 53(2):367–371.

Turner, H., Bailey, T., and Krzanowski, W. (2005). Improved biclustering of microarray data demonstrated through systematic performance tests. *Computational Statistics and Data Analysis*, 48(2005):235–254.

Uhrmacher, A. M., Tyschler, P., and Tyschler, D. (2000). Modeling and simulation of mobile agents. *Future Generation Computer Systems*, 17(2000):107–118.

UNAIDS (2004). 2004 global report on the AIDS epidemic. *Joint United Nations Programme on HIV/AIDS*.

Ushijima, T. (2005). Detection and interpretation of altered methylation patterns in cancer cells. *Nature Reviews Cancer*, 5:223–231.

Ushijima, T. and Sasako, M. (2004). Focus on gastric cancer. *Cancer Cell*, 5:121–125.

Valafar, F. (2002). Pattern recognition techniques in microarray data analysis: A survey. *Annals of the New-York Academy of Sciences*, 980(1):41–64.

Verboven, S., Branden, K. V., and Goos, P. (2007). Sequential imputation for missing values. *Computational Biology and Chemistry*, 31(2007):320–327.

Waddington, C. H. (1949). The genetic control of development. *Symposia of the Society for Experimental Biology*, 2:145–154.

Waki, T., Tamura, G., Tsuchiya, T., Sato, K., Nishizuka, S., and Motoyama, T. (2002). Promoter methylation status of E-cadherin, hMLH1, and p16 genes in nonneoplastic gastric epithelia. *American Journal of Pathology*, 161:399–403.

Walshe, R. (2006). Modelling bacterial growth patterns in the presence of antibiotic. In *Proceedings of 11th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS 2006)*, pages 177–186, IEEE Computer Society.

Wang, H., Wang, W., Yang, J., and Yu, P. S. (2002). Clustering by pattern similarity in large data sets. In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, pages 394–405.

Wiedle, G., Dunon, D., and Imhof, B. A. (2001). Current concepts in lymphocyte homing and recirculation. *Critical Reviews in Clinical Laboratory Sciences*, 38(1):1–31.

Wilkins, J. F. (2005). Genomic imprinting and methylation: epigenetic canalization and conflict. *Trends in Genetics*, 21:356–365.

Witherden, D. A., Kimpton, W. G., Washington, E. A., and Cahill, R. N. P. (1990). Non-random migration of $CD4^+$, $CD8^+$ and $\gamma\delta^+T19^+$ lymphocytes through peripheral lymph nodes. *Immunology*, 70:235–240.

Wooldridge, M. and Jennings, N. (1995). Intelligent agents : Theory and practice. *The Knowledge Engineering Review*, 2(10):115–152.

Wu, Y. and Marsh, J. W. (2003). Gene transcription in HIV infection. *Microbes and Infection*, 5(11):1023–1027.

Wyatt, R. and Sodroski, J. (1998). The HIV-1 envelope glycoproteins: fusogens, antigens, and immunogens. *Science*, 280(5371):1884–1888.

Wyatt, R. and Sodroski, J. (2001). Dendritic cells and transmission of HIV-1. *Trends in Immunology*, 22(4):173–175.

Yan, S. and Sloane, B. F. (2003). Molecular regulation of human cathepsin B: implication in pathologies. *Biologial Chemistry*, 384(6):845–854.

Zhang, C. and Wong, A. K. (1997). A genetic algorithm for multiple molecular sequence alignment. *Computer Application Biosciences*, 13:565–581.

Zhang, S., Yang, J., Wu, Y., and Liu, J. (2005). An enhanced massively multi-agent system for discovering HIV population dynamics. *Lecture Notes in Computer Science*, 3645:988–997.

Zorzenon dos Santos, R. M. and Coutinho, S. (2001). Dynamics of HIV infection: A Cellular Automata approach. *Physical Review Letters*, 87(16).

List of Publications

- Perrin, D. (2006). Agent-Based Modelling of Viral Infection. *ERCIM News*, 64:50–51.

- Perrin, D., Ruskin, H. J., Burns, J. and Crane, M. (2006). An agent-based approach to immune modelling. *Lecture Notes in Computer Science*, 3980:612–621.

- Perrin, D., Ruskin, H. J., and Crane, M. (2006). An agent-based approach to immune modelling: Priming individual response. *Selected proceedings of Third International Conference on Modeling and Simulation (ICMS06)*, pp. 81-86.

- Perrin, D., Ruskin, H. J., and Crane, M. (2006). HIV Modelling - A Parallel Implementation of a Lymph Network. *Selected proceedings of The Third International Conference on Cluster and Grid Computing Systems (CGCS 2006)*, pp. 84-89.

- Perrin, D., Ruskin, H. J., Crane, M., and Walshe, R. (2008). Epigenetic modelling. *ERCIM News*, 72:46.

- Perrin, D., and Burns, J. (2008). Large-scale immune models and visualization. *ERCIM News*, 74:33–34.

- Kerr, G., Perrin, D., Ruskin H. J., and Crane, M. (2008). Empirical vs. Distribution Based Weighting for Gene Expression Graphs. *currently under review*.

- Perrin, D. (2008). In silico Biology: making the most of parallel computing. *currently under review*.

- Perrin, D., Ushijima, T. and Ruskin, H. J. (2008). *Manuscript on the epigenetic model, in preparation*.

Conferences

- Perrin, D., and Ruskin, H. J. (2006). Agent-Based Modelling of Viral Infection: An Introduction (poster). BioNet 2006 conference, Tallaght, Ireland.

- Perrin, D., Ruskin, H. J., Burns, J., and Crane, M. (2006). An agent-based approach to immune modelling. 2006 International Conference on Computational Science and its Applications (ICCSA 2006), Glasgow, UK.

- Perrin, D., Ruskin, H. J., and Crane, M. (2006). HIV Modelling – Parallel Implementation Strategies. Third International Conference on Cluster and Grid Computing Systems (CGCS'06), Venice, Italy.

- Perrin, D., Ruskin, H. J., and Crane, M. (2006). An agent-based approach to immune modelling, Priming individual response. Third International Conference on Modeling and Simulation (ICMS'06), Cairo, Egypt.

- Perrin, D., Duhamel, C., Ruskin, H. J., and Crane, M. (2007). Microarray biclustering: mathematical model and metaheuristic alternatives. International Conference on Computational Methods (ICCM2007), Hiroshima, Japan.

- Perrin, D. (2007). Biocomputation Research: a two-way street between Biology and Computer Science. Invited talk at the Research Institute for Microbial Diseases, Osaka University, Japan.

- Perrin, D., Ruskin, H. J., and Crane, M. (2007). The case for epigenetic modelling (poster). IARC meeting on cancer and epigenetics, International Agency for Research on Cancer, Lyon, France.

- Burns, J., Perrin, D., and Ruskin, H. J. (2008). Network stability and susceptibility in computational epigenetics (poster). Second annual Symposium on Biological Complexity: Genes, Circuits and Behavior, Salk Institute for Biological Studies, La Jolla,

California, USA.

- Perrin, D., Ruskin, H. J. and Crane, M. (2008). Towards epigenetic modelling: an initial map. IET BioSysBio 2008, Synthetic Biology, Systems Biology and Bioinformatics, London, UK.

- Perrin, D., and Ruskin, H. J. (2008). The case for epigenetic modelling (poster). XX International Congress of Genetics, Berlin, Germany.

- Porwal, J., Ruskin, H. J., Perrin, D., and Roche, D. (2008). Microscopic model of epigenetic mechanisms (poster). XX International Congress of Genetics, Berlin, Germany.

- Perrin, D., Ruskin, H. J., and Crane, M. (2008). Inclusion of localised patterns in a large-scale agent-based model of the immune response to HIV infection. Sixteenth Annual International Conference Intelligent Systems for Molecular Biology (ISMB2008), Toronto, Canada.

Glossary

Adaptive immune response: In contrast to the innate response, the specific, or *adaptive*, immune response is based on the accurate recognition of foreign non-self antigens. Antigen-specific response has two arms, namely *cell-mediated* and *antibody-mediated* responses. The latter, also known as the *humoral* response, features B lymphocytes as effector cells, and mainly targets bacterial attacks. Humoral response is characterised by production, by these cells, neutralizing antibodies, following activation by CD4+ T helper cells through release of *interleukin* IL-4. Cell-mediated response is targeted more specifically at viral attacks and takes place in lymph nodes.

Agent: an intelligent agent is a modelling object with specific properties which include autonomy, social behaviour, reactivity, and proactivity.

Agent-based model: a model in which the key abstraction elements are *agents*. When using several agents, such a model is often called a *multi-agent system*.

AIDS: Acquired ImmunoDeficiency Syndrome. Collection of symptoms and infections resulting from the specific damage to the immune system caused by the *human immunodeficiency virus*. This is the last phase of HIV infection progression.

Antigenic determinant: see *epitope*.

Autophagosomes: vesicles which store structures the cell has targeted for destruction through autophagy, (a cellular degradation pathway for the removal of damaged, or superfluous, proteins and cell subunits).

Biclustering: simultaneous clustering of both genes and conditions.

Carboxypeptidases: enzymes which hydrolyzes the carboxy-terminal, (C-terminal), end of a peptide bond. They have diverse functions, (e.g. catabolism, protein maturation).

Chemokines: family of small cytokines. They induce directed *chemotaxis*, (innate movement), in nearby cells, hence their name.

Chromosomes: complex combination, called chromatin, of DNA and proteins, which store all genomic information. Major proteins involved are histones. Nine histones combine to form a nucleosome. The characteristic structure of a nucleosome is that of four pairs of histones forming a core around which about 146 base pairs of DNA is wrapped. This is maintained in place by a linker histone, H1, and repeats over the chromatin every 200 base pairs. The remaining 50 base pair of this repeating unit consists of "linker DNA".

Clustering: grouping of genes, based on their expression under multiple conditions (or over different time-points) or, conversely, grouping of conditions according to expression of several genes.

CpG dinucleotide: a cytosine followed by a guanine in the DNA sequence.

CpG islands: areas with higher proportion of CpG, and formally defined as follows: (i) Length of the considered region is at least 200 base pairs; (ii) GC percentage is greater than 50%, (i.e. more than half of amino-acids are cytosine or guanine); (iii) Observed/expected CpG ratio that is greater than 60%. In humans, these islands are found in or near to 70% of gene promoters.

Cytokines: a category of diverse signalling proteins and glycoproteins, which are essential to cellular communication. Other categories of signalling proteins include hormones and neurotransmitters.

DNA methylation: addition of a methyl group to a DNA strand. In humans, only 1% of DNA bases undergo DNA methylation. In differentiated cells, DNA methylation is typically limited to CpG dinucleotides. Non-CpG methylation can be found in embryonic stem cells. Of particular interest are *CpG islands*. While most CpG are methylated over the genome, these regions have a very distinct pattern: methylation of a CpG island corresponds to silencing of the associated gene. Aberrant changes in CpG island methylation are, therefore, linked with abnormal gene expression.

DNA methyltransferases: enzymes controlling DNA methylation, (e.g. Dnmt1, Dnmt3a and Dnmt3b in mammals).

Dynamic model: conceptual model which describes the states, transitions, events, actions, activities and interactions of the system structures, which characterise system behaviour.

Enzymes: molecules which increase the rates of chemical reactions. This is referred to as a catalytic action.

Epigenetics: study of the heritable changes in gene function that may occur without a change in the DNA sequence. These changes include, for instance, *DNA methylation* and *histone modifications*.

Epitope: a protein site which is recognised by the immune system, (specifically by antibodies, B cells, or T cells). For simplicity, an epitope can be considered as a 3D surface

features of a molecule. These "shapes" fit precisely and thus bind to specific antibodies. Epitopes are also known as *antigenic determinant*.

FLOPS: FLoating Operations Per Second.

Foamy viruses: see *spumaviruses*.

Functional model: conceptual model which describes data flow during system activity, both within and between components.

Gene: basic biological unit of heredity, composed of DNA, (or RNA, for some viruses). They are responsible for the encoding of all biological functions. The total set of genes in an organism is known as its *genome*.

Gene expression: genes have specific functions. Even though each cell contains the whole genetic material, it only uses a fraction of this. The others are *silenced*. These complex dynamics are time dependent and cell-type dependent, and are referred to as gene expression.

Genome: the genome of any living organism is its whole hereditary information. It is encoded in the DNA or, for some viruses such as HIV, the RNA.

Hamming distance: for two strings of equal length, minimum number of substitutions required to change one into the other.

Hemocytoblasts: precursors of lymphoblasts.

Histones: see *chromosomes*.

Histone modification: epigenetic change characterised by the addition, (or removal), of a functional group, (methyl, acetyl, etc.), to specific amino acids of histone proteins. Modifications can occur on tails of histones H3 and H4, and in the core of H2A and H3. Some amino-acids can undergo several successive modifications. Lysine 79 of histone H3 can, for instance, be mono-, di-, or trimethylated. The role of the changes is modification-specific and molecule-specific.

HIV: Human Immunodeficiency Virus. Retrovirus targeting immune cells and using them as hosts. This results in massive depletion of immune cell populations. Infection progression is typically divided in three phases, ending with *AIDS*.

Innate immune response: a non-specific, or *innate*, response is based upon recognition of the pattern of the microbial surface components of the pathogens, rather than by a specific antigenic sequence. Innate response *does not confer* long-lasting immunity to the host, i.e. there is *no memory* of previous responses.

Lamina propria mucosae: "the mucosa's own special layer", in Latin. A thin layer of tissue which, together with the epithelium, constitutes the mucosa. It is often referred to as *lamina propria*.

Lentiviruses: cytopathogens, responsible for slow-progression infections, (hence their name).

Long-term nonprogressors: individuals who have been living with HIV for over 10 years (there is no agreed time span, but authors generally use 10 to 12 years as a threshold), have stable $CD4^+$ counts of 600 or more cells per cubic millimeter of blood, show no sign

of HIV-related diseases, and have not received any antiretroviral therapy.

LTR: long terminal repeat are characteristic of viral genetic material. They ank functional genes, and their main function is to mediate integration of the retroviral DNA into host chromosome. The ve prime, (5), end refers to the end of the DNA, (or RNA), strand that has the fth carbon in the sugar-ring of the (deoxy)ribose at its terminus, (as opposed to the 3 end, which is terminating at the hydroxyl (-OH) group of the third carbon in the sugar-ring, and is also known as the tail end.

Lymphoblasts: immature immune cells, formed in the bone marrow, by differentiation, from precursor hemocytoblasts. These are immature cells, from which prolymphocytes, direct precursors of lymphocytes, are derived.

Lymphopoiesis: generation of lymphocytes. Details of cells and precursors can be found in the glossary.

Macrophages: a type of white blood cell that ingests foreign material. In that sense, they are involved in the non-specific immune response. They are also involved in the specific immune response: they carry the antigen on their surface and present it to T cells.

Microarray: technology used for large-scale transcriptional profiling, through measurement of expression levels of thousands of genes at the same time and under several experimental conditions, (or different time points).

Mitochondria: membrane-enclosed specialized subunits found in most eukaryotic cells. They produce adenosine triphosphate, (ATP), which the cells use as a source of chemical energy.

MPI: Message-Passing Interface, a communications protocol used for parallel implementation of programs. MPI provides support for point-to-point and collective communications, inquiry routines to query the execution environment, as well as constants and data-types.

Multi-agent system: see *agent-based model*.

Neonate: a human infant less than four weeks old.

Non-specific immune response: see *innate immune response*.

Nucleosomes: see *chromosomes*.

Object model: conceptual model which gathers all details on objects within the system, describes their structure, their relations and the operations they support.

Oncoviruses: the largest sub-family of retroviruses. They can induce several types of tumours, e.g carcinoma, lymphoma and leukemia. They have been isolated in humans as soon as early 1980s.

Pathogen: a biological agent which causes disease or illness to its host.

Prolymphocyte: immature immune cells, obtained from lymphoblasts. The last development step in lymphopoiesis, from prolymphocyte to lymphocyte, can take place in two different locations, which will decide the final role of the cells: the prolymphocytes maturing in the bone marrow itself become B lymphocytes, while those maturing in the thymus become T lymphocytes.

Rapid progressors: individuals who progress to AIDS within four years of HIV infection.

Reverse transcriptase: enzyme which transcribes single-stranded RNA into double-stranded DNA. This process is the *reverse* of the normal transcription, which corresponds to the synthesis of RNA from DNA. These enzymes are also known as RNA-dependent DNA polymerases.

RNA-dependent DNA polymerase: see *reverse transcriptase*.

Specific immune response: see *adaptive immune response*.

Spumaviruses: non-pathogens viruses, also known as foamy viruses. They are mainly prevalent in non-human primates, and were first described in the early 1950s. They are easily isolated, thanks to the characteristic foam-like effect they induce.